

July 1983

AN EMAP PUBLICATION

75p

Electronics & computing

MONTHLY

Britain's First Electronics & Computer Applications Magazine

taking the lid off
SORD's M5



**FLOPPY DISC
CARD —**
for our Hi-Res computer

**SEQUENCER
INTERFACE —**
electronic music with the BBC

**SPECTRUM PRINTER
INTERFACE —**
Centronics parallel design



**PLUS
WEATHER SATELLITE DECODER**

Electronics &
Computing Monthly
Scriptor Court,
155 Farringdon Road,
London,
EC1R 3AD

EDITORIAL
01-833-0846

Editor
Gary Evans
Editorial Assistant
Ann Houghton

Administration
Liz Gregory

ADVERTISING
01-833-0531

Manager
Claire Fullerton

Deputy Manager
Alun Evans

PRODUCTION
01-833-0846

Design
Terry Gibbins
Make-up
Time Graphics

Publisher
David Raven

Publishing Director
Peter Peskett

Distribution
EMAP
National Publications

Published By
EMAP Business And
Computer Publications

Printed by
EMLP Peterborough

Subscriptions
Electronics &
Computing Monthly,
(Subscription
Department),
Competition House,
Farmdon Road,
Market Harborough,
Leicestershire.

ABC

MEMBER OF THE AUDIT
BUREAU OF CIRCULATION

Vol 3

IN THIS ISSUE

Issue 7

Weather Satellite Decoder 20 BBC Synthesiser Interface 53

Full details of the hardware and software that will turn the BBC model B into a weather satellite display terminal.

A straightforward design that will appeal to the musically inclined. The project allows the BBC micro to drive any synth with a standard 1 volt per octave input.

Understanding Digital Electronics 27 Computer Brain 58

Part five of the series takes a look at the workings of the major types of memory devices.

Mike James with the last but one in his series on artificial intelligence. This month he deals with fuzzy thinking.

BBC Word Processor 32 Floppy Disc Interface 63

The Wordwise package designed for the BBC micro is put through its paces.

The latest card for our Hi-Res Graphics Computer project. This brings the FLEX OS to life and is the last 'major' component of the system.

Power Control for Micros 36 Sord M5 Review 70

In part two of this project we present PCB designs for the three interfaces described last month.

We take an in-depth look at the latest in the long line of low-cost micro computers aimed at the home/business market.

Spectrum Printer Interface 39 Spectrum Video Processor 76

This project enables the Spectrum to drive the majority of low cost parallel printers, and so provide a high quality output from the computer.

This month we describe the software that allows last month's interface to provide a high resolution image.

Best Straight Line 47 IC Update 83

Paul Beverley teams up with Steve Dowson to present a program to draw the bestfit line through a series of points.

Details of the pin-outs of devices in the popular 74xx series of TTL ICs.

Comment	9
Next Month	9
News & Views	10
New computer products	13
Subscriptions	25
Readers letters	75

Our regular features 'Understanding Your Computer' and 'Softly Softly' have been held over due to lack of space.

Electronics & Computing Monthly is normally published on the 13th day of each month

© copyright EMAP Business & Computer Publications Limited 1983. Reasonable care is taken to avoid errors in this magazine however, no liability is accepted for any mistakes which may occur. No material in this publication may be reproduced in any way without the written consent of the publishers. Subscription rates: UK £9.50 incl. post. For overseas rates apply to Subscription Dept., Competition House, Farmdon Road, Market Harborough, Leics.

EDITORIAL

The micro computer trade is one area of the consumer market place in which we here in Britain have managed to hold our own. Clive Sinclair, from his days at Science of Cambridge with the MK14, through the ZX80/81 to the Spectrum has maintained a healthy share of the developing trade. Chris Curry of Acorn, Chris incidentally also cut his teeth on the MK14, has the BBC machine to his credit. The Spectrum and the BBC micro have between them a large share of the low-cost end of the micro market.

The Americans have also managed to achieve significant sales in this area, from the pioneering days of Commodore's first PET and Tandy's early TRS-80 with the, still current, VIC-20 and continued, if steady, sales of the Atari range.

Recently a new factor has entered the equation with the emergence of a string of new British companies coming on stream with the likes of the Oric, Dragon and Lynx. It's with a certain smug satisfaction that these British products offer a performance/price ratio that can give designs from any source a run for their money.

If we're not careful though – all this could change.

History Repeats Itself?

Those of us with long enough memories can cast our minds back to the late fifties, early sixties and to the fledgling hi-fi industry. It was radiograms and 'portable' record players that were then the order of the day. They went under the banners of names like Ferguson, Thorn and Roberts – all good British Companies. Then the Japanese set their sights on these shores and, within a comparatively short period of time, had managed to reduce the British industry to badge engineering in the mass market and to the production of high quality and cost, low volume products.

The battle, such as it was, was fought on grounds of price, performance and reliability.

The Far East has been strangely quiet in the £50 - £400 computer market to date but all this is about to change. This issue of *E&CM* carries news of the Sord M5, a new low-cost machine from Sharp and an advert for the Comex. Just three machines and there are plenty more due for release in the near future.

In the areas of price and performance, these imports as yet have nothing much more to offer than their home grown counter parts. In terms of reliability and delivery though they are set to make British computers look rather poor relations. No longer will British companies be able to indulge in the luxury of launching under developed products and treating delivery promises with a fair degree of scorn.

Let's hope that the potential threat posed by the Far East does not provoke demands for trade barriers but instead spurs the British into producing quality, high performance products *on time*.

Price Wars

The price war that has broken out in the £100/£200 range of computers has had one welcome side effect. The price of the ZX81 has fallen to a remarkable, all time low level. At its present price, it is an ideal choice for many dedicated applications. Our Central Heating Controller project has shown the way but there are a host of other applications to which the ZX81's power could be applied. Granted the computer does not feature the most generous of Input/Output facilities, but one can't have everything.

So if you've ever wanted to design a sophisticated burglar alarm, train controller, hi-fi control unit, juke box etc., why not sit down and design around a ZX81, some ROM, I/O and a minimum amount of discrete circuitry.

Let us know how you get on.

GARY EVANS

Look out for our August Issue

On Sale July 13th Ode To Joy ORIC Special

Next month we feature an exhaustive review of the software released to date **PLUS** an interface that greatly increases the I/O capabilities of the computer.

PLUS

The articles on the BBC assembler and a digitizer for the machine, held over this month due to pressure on space, WILL appear next month. We'll also conclude our look at Word Processing on the BBC with a look at VIEW from Acornsoft.

ZX Sound Boards

Generating music with your micro is all the vogue at the moment. While other magazines have gone into the theory of music synthesis, *E&CM* gets down to the nitty gritty of the subject with designs for Spectrum and ZX81, add-on boards.

These articles also explain the detailed workings of the popular AY-3-8910 Programmable Sound Generator around which both designs are based.

You may have seen similar designs before but they will not have been presented in our technically biased fashion.



480Z Price Drops

Research Machines have been doing rather well with their LINK 480Z Micro. So well in fact that they have been able to drop the price of the machine and their network products by up to 13%.

In addition RM are offering a 10% discount on the new prices to schools and colleges. That takes the price of an L2 system down to £483 if LEA quantity discounts are taken into account.

RM are at:-
Mill Street
Oxford
OX2 UBW



On the grapevine

Sharp are poised to break into the low cost home/business market with a machine designated the MZ731. The computer is Z80 based and thus capable of running CPM packages. It offers sound hi-res graphics (in colour) and has 64K of RAM. The version we've seen features an integral (ALPS?) colour printer/plotter and cassette deck. It's possible though that the version offered in the UK will consist of the main unit with peripherals available as extras.

The predicted price for MZ731 is around £250 and it's possible that the machine will be on sale as early as August.

Oric Soft

The first batch of Oric software is now available from dealers. The five packages have been specially commissioned and are a mixture of games, simulation, and a more serious title.

The Multigames tape offers a set of five 'family games' while the Flight tape provides an aircraft landing simulation. Zodiac is a rare playing adventure game that retails for £7.95 as do the other games.

Oric chess will sell at £9.99 and features five levels of difficulty. Finally Oric base, again at £9.99 is a Data Management program, complete with a manual, that incorporates an easy to learn Query language.

BBC WP

Stable software are offering a full screen editor/wordprocessor under the name of SCRED.

A major feature of SCRED is that it may be used both for basic programs and for letters or documents.

Existing Basic may be edited by the package and stored in the normal compressed format. Documents can be saved and retrieved by SCRED in a different format that allows a variety of formatting controls, understood by the SCRED PRINT COMMAND, to be saved with the text.

The package is said to be user-friendly as are the people at Stable Software. Their address for any further information is:-

Millsail House
Great Saxham
Bury St. Edmunds

Computing in Cambridge

A new firm, Cambridge Computing, are launching a unique joystick interface that is compatible with both ZX81 and Spectrum.

The interface differs from the majority of those on the market in that it will work with all software irrespective of whether it is written for joysticks or not.

For more details contact:-
Cambridge Computing
1 Benson Street
Cambridge
Telephone 522905

Club Call Echo

The response to last month's Computer Club Call was encouraging. We'll start a regular listing of Clubs next month and we've still got room to mention any Club that cares to drop us a line with details of their up and coming events.

Brief details to the editor at:
Electronics & Computing
Scriptor Court
155 Farringdon Road
London
EC1R 3AD

marking the envelope 'Computer Club Call'.

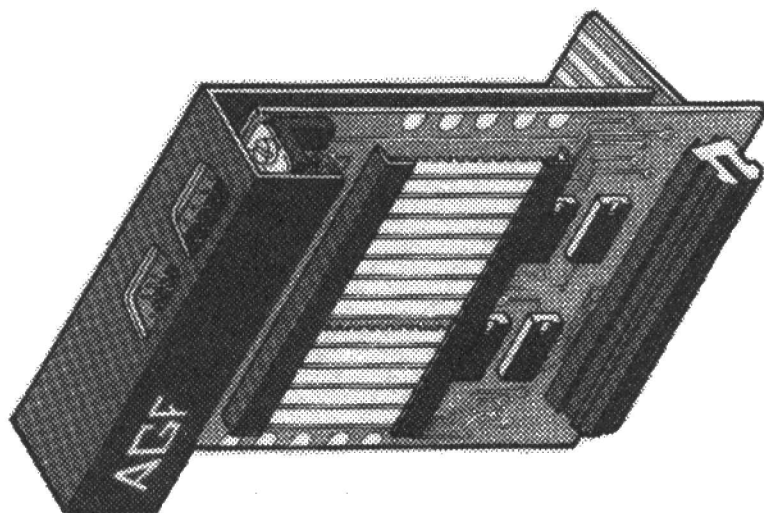
Maplin and Heathkit

The range of Heathkit products will in future be distributed by Maplin Electronics.

Maplin, with its established retail and mail order operation were considered an ideal company to take on the Heathkit business when Zenith Data Systems (Heath's parent company) decided to close down its own UK marketing operation.

At present the full range of Heathkits are not available but, by November, Maplin will be selling the complete selection of products.

Maplin are at:-
PO Box 3
Rayleigh
Essex
SS6 8LR



More Joy

August this year will see the launch of a programmer joystick interface from AGF hardware. The interface is designed for the Sinclair Spectrum and ZX81 and will offer joystick compatability with all games software.

The interface can be used with one or two Atari-compatible joysticks. The design allows simulation of any key by one of the four joystick directions and the fire button. The interface is supplied with clip-on programming leads which are configured on the vertical busbars according to a chart showing each key's position.

Full instructions will be supplied with the interface together with a demonstration cassette with a program that allows high resolution drawings to be produced under joystick control.

The price of the interface is to be £32.95 plus £1.00 p&p.

AGF Hardware are at:-
26 Van Gogh Place
Bognor Regis
West Sussex
PO22 9BY

Econet for Z80's

The Econet local area network developed by Acorn has until now suffered from a lack of software for non-Acorn products which are all 6502 based. SJ research have been

working on an interface card which will overcome the difficulty by allowing Z80 based systems to join the Econet network. The card is designed for research machines 380Z, popular among educational establishments, and gives the computer access to network facilities such as Five Servers (remote disks) and Print Servers.

The basic price of the interface is £153 plus VAT.

Under development at present is a similar S-100 system to Econet interface.

Further information from:-
SJ Research
108 Mill Road
Cambridge
CB1 2DB

Let battle commence

W. H. Smith look like they've started a price war with their £10.00 ZX81 price cut that pre-empted Sinclair's own price reduction by a full week.

The move was seen as an attempt to dump ZX81 stocks before being left with an unsellable product in view of the co-incidental reductions on the Spectrum's price. It is said Smiths had 20,000 81's on their hands and the thought of the £100 Spectrum provoked them into the price cut move. The rest of the 'established dealer network' were none too pleased.

Cut and thrust

Texas and Commodore are not letting the Sinclair price cuts go unchallenged.

TI are to package their 99/4A, together with vouchers entitling the purchaser to a pair of joysticks plus a Beginners Tutorial Program and Connect Four, at a price of £149.95.

This puts the price of the computer itself at around £100.

Commodore are taking the same route with a deal that reduces the price of the VIC-20 to £85. The customer will have to pay £139.95 though. It is the inclusion of the special VIC cassette deck, a teach-yourself Basic pack and four free games or learning packages, that bring the cost of the computer down to the £85 mark.

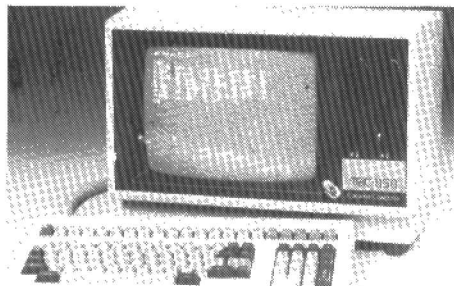
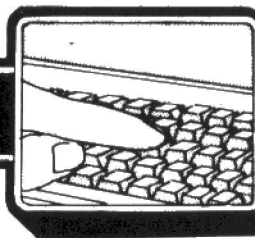
Whether these 'package deals' will be as successful as Sinclair's straightforward price cuts when it comes to shifting stock remains to be seen.



Chewing Gum?

A new magazine for Colour Genie owners avoids the problem of finding a new name for a publication when the permutations of the words computing/micro/what/which/how/when have already been used. Chewing Gum is the name of the newcomer and if you are a Colour Genie user than this is the magazine for you, in addition to Electronics and Computing, that is.

NEW PRODUCTS

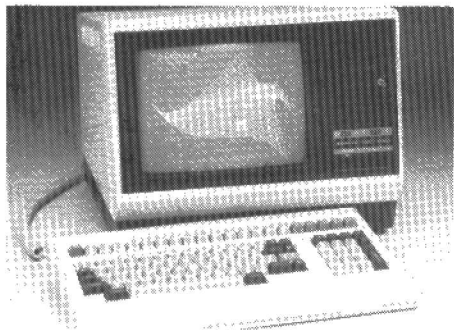


Sanyo Newcomer

The range of 8-bit machines from Sanyo is further extended with the introduction of two new machines, the MBC 1150 and the MBC 1250.

The 1150 replaces the MBC 1000 and offers two 5.25" drives as opposed to the single drive of its predecessors. Based on two Z80A's with no wait modes. The computer offers 64K of RAM, 2K of video RAM plus the 320K bytes of storage of the disk drives.

The computer's screen is configured as 80 x 25 lines and the detachable keyboard has 15 programmable function keys.



The MBC1250 is a new departure for Sanyo - A business computer with the graphic capability afforded by a 640 x 400 dot matrix display. An auxiliary Z80A provides a speedy execution of graphics functions. The machine again features 64K of RAM and a 4K ROM.

Prices are £1,585 for the 1150 and £1,995 for the 1250. Both prices are exclusive of VAT.

High Quality

Reflex Limited announce the availability of the Electrohome ECM 1302 high-quality, thirteen inch colour video monitor for the first time in the UK.

This monitor has already achieved considerable success in America especially with users of IBM and Apple personal computers.

The ECM 1302 monitor is available with either medium or high resolution, allowing the customer to select a unit that best suits the video performance of the computer. The ECM 1302-1 has a resolution of 370 by 235 pixels whilst the higher performance 1302-2 has a 580 by 235 pixel resolution. Both monitors will display up to twenty-five lines of eighty characters to match the output of most personal computers.

The Electronics colour monitors will be available via the established dealer network. Reflex also have the capacity to modify the monitor, or build special interfaces, to meet the needs of systems manufacturers.

Memory Module

Cambridge Microelectronics have recently introduced two ZX81 add-ons. The first, Promer-81, is an EPROM programmer retailing for the very reasonable price of £19.95. The Promer is designed for use with 2516, 2532, 2716 and 2732 Eproms. All the standard programmer functions of CHECK, SPECIFY, READ, PROGRAM and VERIFY are provided with various safety features, e.g. checking the Vpp status before programming, also a part of the specification.

The control program, supplied on tape, is menu driven and guides the user with a series of on-screen prompts.

Power for the Promer-81 is supplied by four PP3 batteries. The unit is supplied fully assembled and tested.

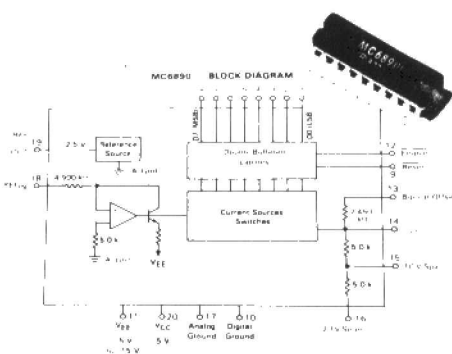
The second new design from Cambridge Microelectronics is designated the Cramic-81. This is a 16K rampack based on CMOS memory devices. These memories feature a very low current to power them in fact with the Lithium battery supplied with the unit, a life of 10 years is predicted.

The unit can be used in tandem with standard rampacks and in this case hardware/software switching allows the user to select either 16K

block of memory.

For further details on both products contact:-

Cambridge Microelectronics
1 Milton Road
Cambridge
CB4 1UY
Telephone (0223) 314814



MPU D/A

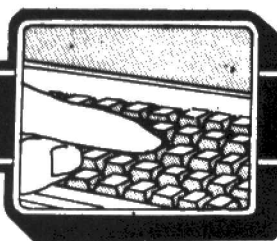
The Motorola MC 6890 is a new bus compatible 8-bit D/A converter that boasts an impressive array of features. The device is capable of interfacing directly with 8-bit microcomputers, and is a monolithic converter containing master/slave registers to prevent data transmissions during active enable; a laser-trimmed, low-TC, 2V5 precision bandgap reference; and high stability, laser trimmed, thin film resistors for both reference input and output span, the bipolar control.

A reset pin provides for overriding stored data and forcing output current to zero.

Other features of MC6890 include direct data bus link with all popular TTL level MPU's; +112 LSB nonlinearity over temperature; fast settling time, typical 200ns; low power requirement, typically 90mW; and +5V and -5V to -15V supplies.

For full technical specifications and availability contact:-

Axiom Electronics
Turnpike Road
Cressex Industrial Estate
High Wycombe
Bucks
HP12 3N



Spectronics

Elsewhere in this issue, the more enterprising of our readers find the design for a DIY Centronics interface for the Spectrum. For those of you who would prefer a ready made piece of equipment, a new product from Kempston (Micro) Electronics should be of interest.

A major feature of the unit is the recognition of LLIST and LPRINT. This allows programs to be listed directly from the Spectrum and also allows printout direct from listings (BASIC only) without the need of special user calls. It is possible to send out control codes to the printer giving the facility of different characters i.e. condensed, expanded, etc. The interface is supplied cased and ready to use by simply plugging directly into any Centronics type printer i.e. all Epsoms including MX-80F/T111, Seikosha 100 A, OK1 Microline 80 etc. and also includes driving software which allows up to 128 characters per line. (Depending on printer type).

The interface complete with connecting lead will be priced at £45.00 including VAT (Mail order £1.00 p&p) which includes a 12 month guarantee. Kempston Electronics are at:-

180a Bedford Road
Kempston
Bedford
MK42 8BL

Speaker's Corner

Cyber Robotics are now able to supply the Votrax speech synthesiser unit.

The most interesting feature of the unit is its ability to receive serial or parallel data which is automatically translated into electronic speech by an onboard, microprocessor based, text-to-speech algorithm, which is accurate for 95% of English words. Text can be keyboard or computer generated – the unit's vocabulary is almost unlimited.

The criticism of some units has been the robotic-like sound produced, this has been considerably reduced in the PSS with the incorporation of programmable inflection and amplitude which permits the user to



create more realistic speech. The addition of music with ADSR – programmable attack/decay/sustain/release – which is the way musical instruments behave, permits great accuracy with creativity.

The system also incorporates a built-in clock with programmable spoken alarms.

The unit is small (12.3" x 5.2" x 3.1") and weighs in at 2.6lbs. A comprehensive 88 page manual covers all of the questions in detail.

The price is £375 plus VAT with UK power supply.

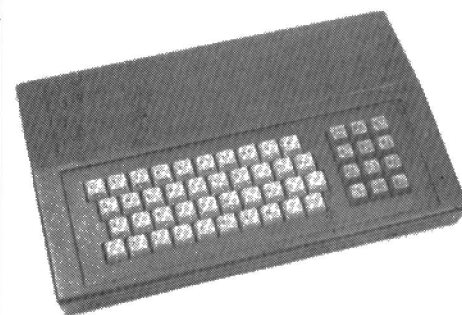
Further information from:-
Richard Springett
Cyber Robotics Ltd.
61 Ditton Walk
Cambridge
CB5 8QD

Print Serve

Having saved up for a BBC model B it's likely that it will be a few months before you can afford a printer. In the meantime, to avoid the frustration of being unable to list your programs, it will be possible to obtain a printed listing from your cassette tapes by courtesy of Beebprint. The company offer a return of post service and their charges are quite reasonable.

A listing for a program with a maximum block count of £20 is £1.95 and the excess charge to larger programs is 20p per 9 blocks.

More details from:
19 Orchard Way
Hurstpierpoint
West Sussex
BN6 9UB
Telephone 0273 833397



Quality Keyboard

DK Tronics have announced the launch of a new Spectrum Keyboard. The cased keyboard has 52 keys, 12 of which are used for a numeric pad that, as well as making entry of numeric data far easier, allows cursor control with one hand.

The Spectrum must be removed from its original case and fitted inside the new enclosure – fitted bosses and marked screw holes make this a straightforward operation. There is plenty of room within the case for additional modules, for example a power supply and all connections and the expansion port are available at the rear of the case.

The DK Tronics case is fully compatible with all existing Spectrum add-ons.

Further details from:-
DK Tronics
Shire Hall Industrial Estate
Saffron Walden
Essex
CB11 3AX
Telephone 0799 26350

BBC Weather Satellite Display

Michael Furminger, of Nene College, Northampton, describes a system for displaying Weather Satellite pictures on the BBC model B computer.

Many schools, colleges and keen amateurs have, over the years, been tracking and reproducing weather pictures transmitted by American NOAA and Russian Meteor satellites. Receiving the transmission causes little problem, but presenting a good image can be very expensive if a facsimile printer is used, or unsatisfactory if an oscilloscope is pressed into service. This article shows how a BBC model B computer (or a model A + 32K memory + 6522 VIA) can be used to present the picture.

The principle of the system is to save the transmissions of the satellite together with a timing pulse on to a stereo tape recorder. The recording is played through an analogue to digital converter (ADC) into the computer. Line synchronisation is achieved using a clock pulse recorded with the signal.

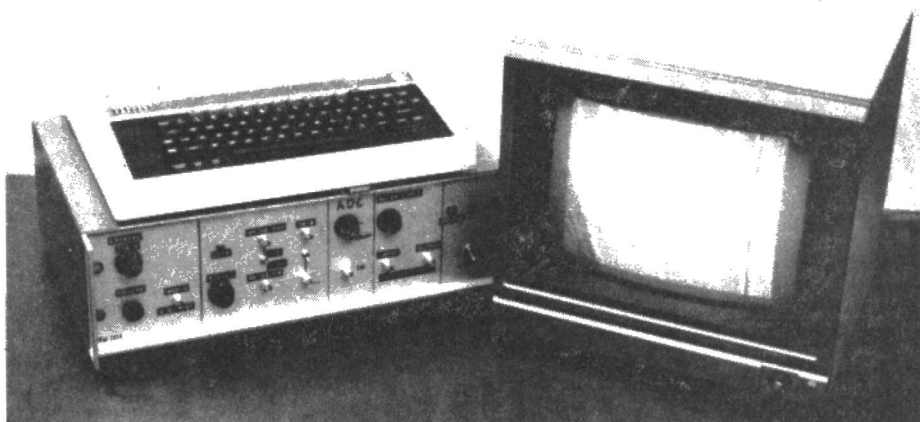
The satellites chosen to provide pictures are the American NOAA6 and NOAA7, both of which are in low earth polar orbit. NOAA6 passes at around 08.00 and 20.00 hours, the frequency for the low resolution pictures being 137.62 MHz. NOAA7 orbits such that a pass is around 12.30 hours and transmits at 137.5 MHz. Both satellites have a period of about 1 hour 41 minutes and precess by about 25°W. This means that three passes can be recorded from NOAA7's mid-day passes. Once a good pass is received, the time for the pass on the next day is given by the equation:

$$\text{Time of pass} = 1.69 * \text{INT}(14.22 * X) + \text{BT} - 24 * X$$

where X = number of days
BT = Base time

Scanning Techniques

The pictures are produced by scanning Radiometers which are then transmitted line by line in a continuous stream. The signal transmitted in the 137 MHz band is FM and is produced by an AM modulated 2400 Hz carrier.



The Hardware

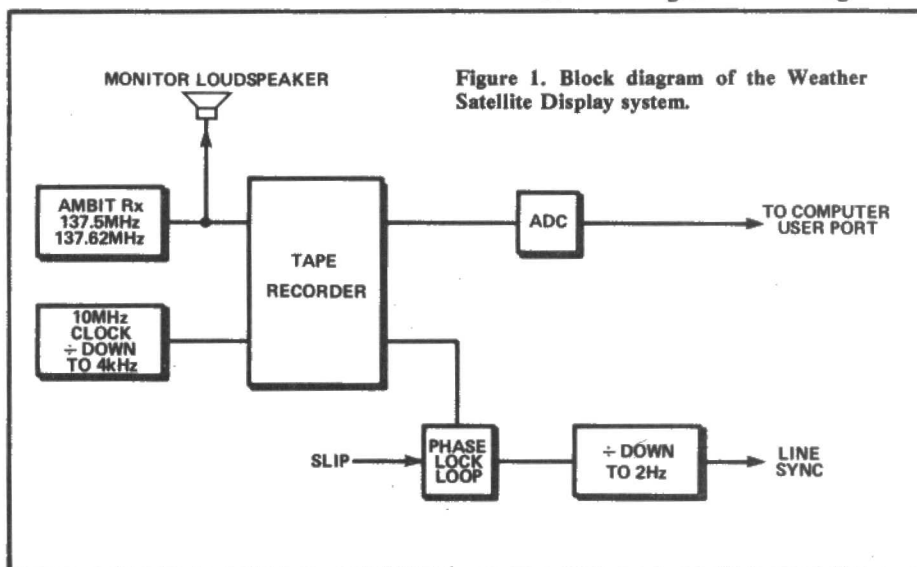
The signal transmitted by the satellites is in the 137 MHz FM satellite band. The receiver is an NBFM Kit 96640 produced by Ambit International modified to accept a wider band width of 50 KHz.

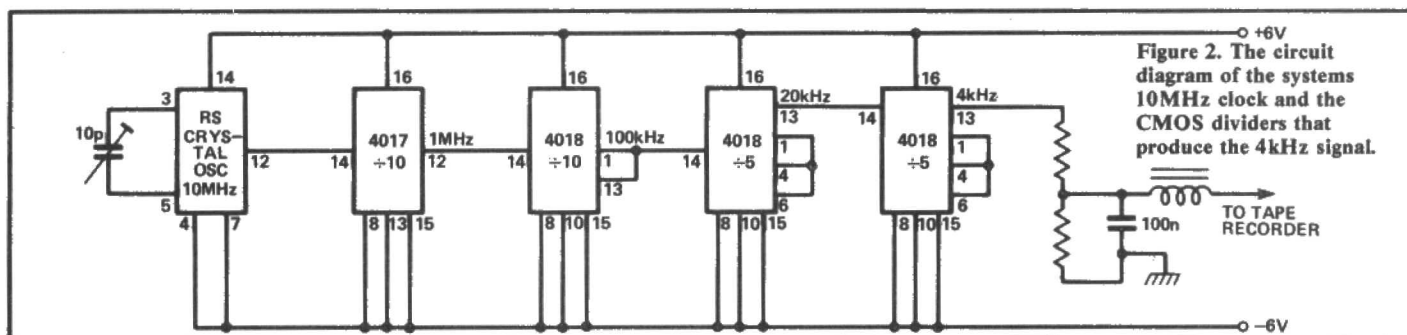
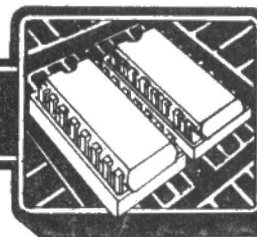
A ground plane crossed dipole aerial is used which enables a 20 minute satellite pass to be received without moving the aerial.

Each line takes 0.5 seconds and is made up of synchronisation pulses and two pictures, one visible and the other Infra Red.

The picture built up by the receiver will be two pictures side by side. On a clear sunny day the ground can be seen in the visible picture and on a clear night in the IR picture.

The signal is monitored with a small loud-speaker driven by an LM 380 amplifier. When the signal is received it is easily identified, since it sounds like a clock ticking with a half second period. This signal can be directly processed by the ADC circuit or stored on a tape recorder. The signal at this stage is an





AM 2400 Hz signal. The 2400 Hz carrier could be used to synchronise the picture but this is unadvisable since satellite signals are prone to fade and interference.

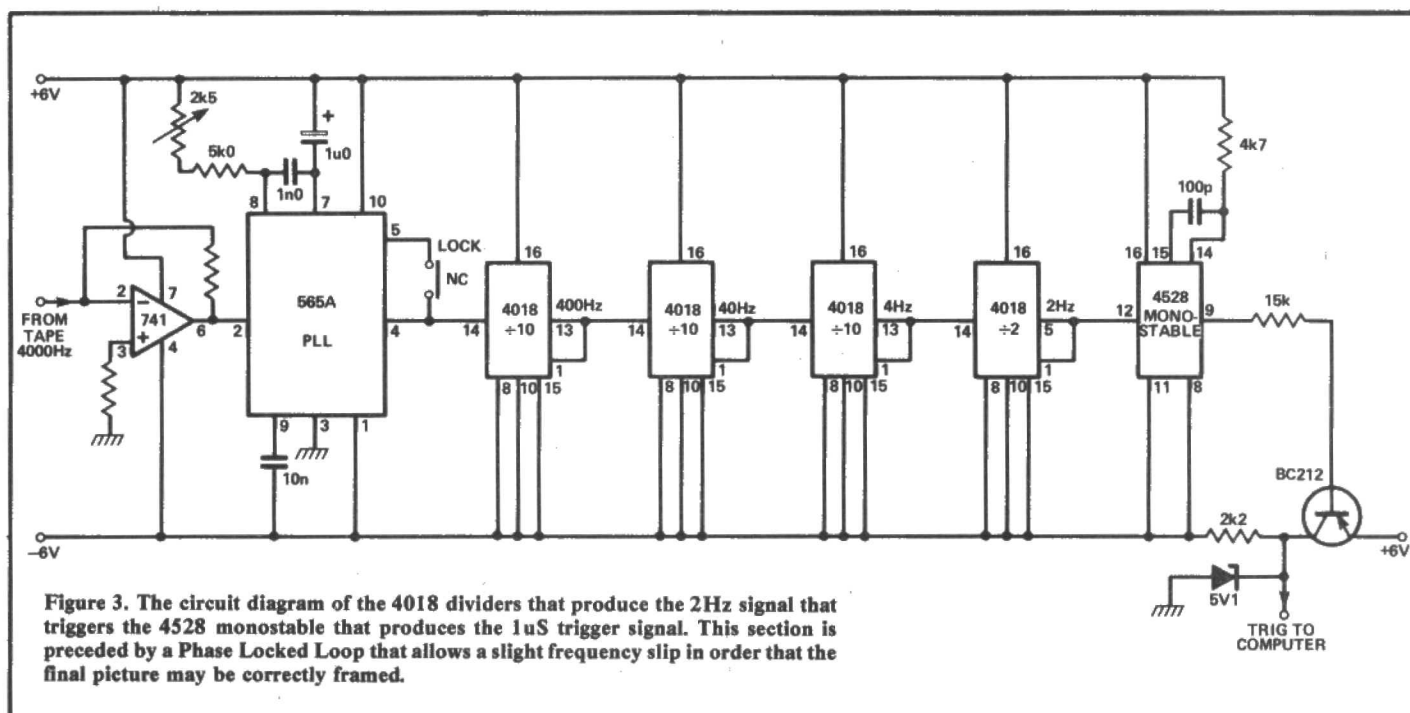
To achieve synchronisation of the picture a 10 MHz crystal oscillator is divided down to a suitable clock rate of 4000 Hz which is filtered by an inductor capacitor network and recorded with the satellite signal onto the other channel of

the tape recorder. The signal filtering was necessary since recording square waves onto magnetic tape produces severe distortion. This is avoided if the square wave is 'rounded' by the filter.

The clock uses an RS crystal oscillator trimmed with a variable 10pF capacitor. This is then divided by a series of CMOS dividers. The clock signal could be replaced by using the BBC 1 MHz bus clock divided by suitable TTL or CMOS

circuits.

The signal from the 4000 Hz stage, either direct or from the tape recorder, is then fed into a Phase Lock Loop (PLL) NE565A. The object of this stage is to enable a slight slip in frequency which allows the received picture to be framed properly. The PLL is set to a slightly lower frequency than 4000 Hz so that when pins 4 and 5 are disconnected there will be a slow picture slip to the left.



When pins 4 and 5 are connected the PLL will lock into the clock frequency from the 4000 Hz stage. This clock is then divided down to 2 Hz by a series of CMOS 4018 dividers and finally a short pulse is produced by the 4528 monostable. The pulse width is set by the 4k7 resistor and 100pF capacitor. This is not critical but is best left at approximately

1 us. Since the clock is CMOS it may be powered by batteries or by a suitable +6; 0, -6 volt supply.

The analogue to digital converter (ADC) interface is based around the Ferranti ZN447 ADC. This is a very fast ADC with conversions in 9 μ s. The BBC computer has a built-in ADC but conversion time is only 10ms, which is

adequate for most purposes, but not fast enough for this job.

The amplitude modulated signal requires filtering rectification and amplifying before the ADC can convert the signal. This is done with a 3140E CMOS Op Amp with the negative power supply pinned to earth. The 100k feedback resistor allows for a wide range

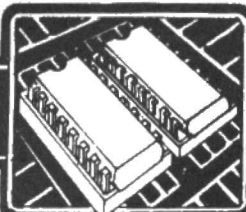


LIST

```

720BEQ START
730.L2 INC HORZH
740LDA HORZH
750CMP #5
760BNE L4
770LDA #129:L0X #0:L0Y #0:JSR OSBYTE
780CPY #0:BEQ R1
790CPY #41B:BEQ ESC
800LDA #0
810STA HORZL:STA HORZH
820L0Y #4
830.Q1 DEC VERTL
840BEQ Q2
850.Q3 DEY
860BNE Q1
870JMP NL
880.Q2 DEC VERTH
890LDA VERTH
900CMP #255
910BNE Q3
920CLI
930RTS
940.R1 LDA #16:JSR OSWRCH
950JMP RESTART
960.ESC LDA #126:JSR OSBYTE
970CLI:RTS
980J
990NEXT
000REM PICTURE CHOICE
010MODE1
020PROCINFO
030MODE2
040VDU23,1,0;0;0;0;
050REPEAT PROCPIC
060UNTIL FALSE
070VDU23,1,1;0;0;0;
080END
090DEF PROCINFO
100COLOUR131:COLOUR0:CLS
110PRINT TAB(1,5):"SLOW SCAN PICTURE
M.F.Furminger 2/83)"
120PRINT TAB(5,10):"COLOUR CHOICE MODE2"
130PRINT
140PRINT"BLACK:RED:GREEN:YELLOW:BLUE:MAGENTA:
CYAN:WHITE ENTER 1"
150PRINT
160PRINT"BLACK:MAGENTA:RED:YELLOW:GREEN:BLUE:
CYAN:WHITE ENTER 2"
170R=GET
180IF R=&31 THEN 1200
190IF R=&32 THEN PROCCOL ELSE1170
1200PRINT TAB(20,16):CHR$(R)
210PRINT TAB(5,20):"TO START PRESS 'SPACE'"
220PRINT
230PRINT"TO RESTART AT TOP LEFT HOLD DOWN 'SPACE'"
240PRINT TAB(5):"TO ESCAPE PRESS 'ESCAPE'"
250PRINT TAB(5,28):"WHEN PICTURE IS COMPLETE"
260PRINT TAB(5,30):"PRESS 'SPACE' TO CONTINUE"
270REPEAT UNTIL GET=&32
280ENDPROC
290DEF PROCPIC
300CLS:CLS
310CALL PICTURE
320REPEAT UNTIL GET=&32
330ENDPROC
340DEF PROCCOL
350VDU19,1,5;0;
360VDU19,2,1;0;
370VDU19,4,2;0;
380VDU19,5,4;0;
390ENDPROC

```



The Mode 1 Program

```

10REM OS1.2 MODE1 PICTURE
20REM SET UP PORT
30MODE1
40PA=&FE61
50PB=&FE60
60DRB=&FE62
70DRA=&FE63
80PCR=&FE6C
90ACR=&FE6B
100FR=&FE6D
110OSWRCH=&FFEE
120OSBYTE=&FFF4
130VERTL=&70:VERTH=&71
140HORZL=&72:HORZH=&73
150?DDR=0:REM INPUT
160?DRA=255:REM OUT
170?PCR=2:REM PB LATCH ENABLE
180?PCR=16:REM CB1 LOW TO HIGH.CA1 H-L
190DIM PICTURE 200
200FORZ%=0TO2 STEP 2
210P%=PICTURE
220CPTZ%
225SEI:CLD
230.RESTART LDA #15:LDX #1:JSR OSBYTE
240LDA #0
250STA HORZL:STA HORZH
260LDA #&FF:STA VERTL
270LDA #3:STA VERTH
280.L LDA FR:WAIT NEW LINE
290AND #2
300BEQ L
310STA PANCLEAR FLAG
320.NL LDA FR:WAIT NEW LINE
330AND #2
340BEQ NL
350STA PANCLEAR FLAG
360.START LDA #208:CB2 LOW THEN HIGH
370STA PCR
380LDA #240:STA PCR
390.LOOP LDA FR:HANDSHAKE ADC
400AND #16
410BEQ LOOP
420LDA #18:COL3:PB/64
430JSR OSWRCH
440LDA #0
450JSR OSWRCH
460LDA PB
470LSR A:LSR A:LSR A:LSR A:LSR A:LSR A
480JSR OSWRCH
490LDA #25:PLOT69,HORZL,H,VERTL,H
500JSR OSWRCH
510LDA #69
520JSR OSWRCH
530LDA HORZL
540JSR OSWRCH
550LDA HORZH
560JSR OSWRCH
570LDA VERTL
580JSR OSWRCH
590LDA VERTH
600JSR OSWRCH
610LDX #40:DELAY
620.D DEX
630BNE D
640LDX #4
650.L1 INC HORZL
660BEQ L2
670.L4 DEX
680BNE L1
690BEQ START

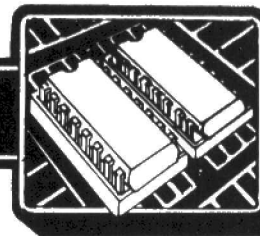
```

```

700.L2 INC HORZH
710LDA HORZH
720CMP #5
730BNE L4
740LDA #129:LDX #0:LDY #0:JSR OSBYTE
750CPY #0:BEQ R1
760CPY #&1B:BEQ ESC
770LDA #0
780STA HORZL:STA HORZH
790LDY #4
800.Q1 DEC VERTL
810BEQ Q2
820.Q3 DEY
830BNE Q1
840JMP NL
850.Q2 DEC VERTH
860LDA VERTH
870CMP #255
880BNE Q3
890CLI
900RTS
910.R1 LDA #16:JSR OSWRCH
920JMP RESTART
930.ESC LDA #126:JSR OSBYTE
940CLI:RTS
950J
960NEXT
970REM PICTURE CHOICE
980PROCINFO
990VDU23,1,0;0;0;0;
1000REPEAT PROCPIC
1010UNTIL FALSE
1020VDU23,1,1;0;0;0;
1030END
1040DEF PROCINFO
1050COLOUR131:COLOUR0:CLS
1060PRINT TAB(1,5):"SLOW SCAN PICTURE
(M.F.Furminger 2/83)"
1070PRINT TAB(5,10):"COLOUR CHOICE MODE1"
1080PRINT
1090PRINT"BLACK:RED:YELLOW:WHITE ENTER '1'
BLACK:RED:GREEN:BLUE ENTER '2'

1100R=GET
1110IF R=&31 THEN 1130
1120IF R=&32 THEN PROCCOL ELSE1100
1130PRINT TAB(35,12):CHR$(R)
1140PRINT TAB(5,20):"TO START PRESS 'SPACE'"
1150PRINT
1160PRINT"TO RESTART AT TOP LEFT HOLD DOWN 'SPACE'"
1170PRINT TAB(5):"TO ESCAPE PRESS 'ESCAPE'"
1180PRINT TAB(5,28):"WHEN PICTURE IS COMPLETE"
1190PRINT TAB(5,30):"PRESS 'SPACE' TO CONTINUE"
1200REPEAT UNTIL GET=32
1210ENDPROC
1220DEF PROCPIC
1230CLS:CLG
1240CALL PICTURE
1250REPEAT UNTIL GET=32
1260ENDPROC
1270DEF PROCCOL
1280VDU19,2,2;0;
1290VDU19,3,4;0;
1300ENDPROC

```



Lines 320-350. Waits for the next new line to ensure picture synchronisation.

Lines 360-410. The ADC routine uses the CB2 to start convert and CB1 to indicate end of convert.

Lines 420-480. This is the GCOL statement to set colour of pixel. Each LSR A in line 470 provides an integer divide by 2.

Lines 490-600. This is the PLOT 69, X, Y routine using high and low bytes of each address.

Lines 610-630. This is a delay to set a screen full time to about 125 seconds. 128 seconds being the actual time of a screen to fill so this effectively stretches each line a little so that uninteresting timing marks can be ignored. Line 610 may be increased up to #50.

Lines 640-730. Increment the Horizontal scan.

Lines 740-760. Enable a restart or escape at the end of each line.

Lines 770 to 880. Decrements the Vertical counter.

Line 890. Clears the interrupt mask set in line 225. This ensures exact timing along each line.

Lines 910 to 920. Perform a CLS to clear screen.

Finally lines 930 to 940 acknowledge an escape condition.

The Basic procedures are very straightforward. PROCCOL uses the VDU 19 statement to change the logical colours. Any four of eight colours may be used.

The Mode 2 program is basically the same the only differences being in the

horizontal counters and in the delay loop, this being very much longer.

Conclusions

The overall result is a cloud cover picture in visible and Infra red which is of reasonable resolution but with the distinct advantage of colour, the colour enabling easy analysis of the pictures.

The ADC and the computer program together make up a useful package for anyone interested in receiving slow scan TV pictures from any source.

Grateful thanks are expressed to: Dr. Trevor Harris of Leicester University Physics Department for the clock circuit design. Mr. Hubert Luxton for the alignment of the Ambit Receiver.

And Nene College for development facilities.

E&CM

MAKE SURE OF GETTING YOUR COPY! HAVE IT DELIVERED EVERY MONTH



"Electronics & Computing Monthly" has become required reading for thousands of Engineers, Technicians, Managers, Company Directors, Teachers, University Lecturers and Students plus many other electronics and computing enthusiasts.

The magazine has a unique blend of theory, projects, software and technical features supported by technology updates — science and general features on a wide variety of high technology topics.

If science, electronics and computing are your areas of interest then you can guarantee a good read in "Electronics & Computing Monthly". Special product features each month will combine interesting articles on a particular product theme with special promotions by advertisers. Special products on offer will beat all others on price. Scopes, digital multimeters and home computers are among the variety of goods which will be available at new low prices throughout the year.

SEND TO

ELECTRONICS & COMPUTING MONTHLY

SUBSCRIPTIONS DEPT
COMPETITION HOUSE
FARMDON ROAD
MARKET HARBOROUGH
LEICESTERSHIRE

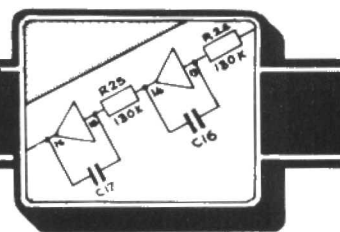
Please send Electronics & Computing Monthly for the next 12 issues to commence from Issue.

I have enclosed a cheque/Postal order for **£9.50** (UK only)
£13.95 (Overseas)

Name

Address

Town



MEMORIES

In part five of the series "Understanding Digital Electronics", J. Oliver Linton explains the internal workings of semiconductor memory devices.

The terms static or dynamic RAM and EPROM are familiar enough but the internal workings of these devices are probably more of a mystery. An understanding of the topics covered in earlier parts of this series will, however, enable the basic principles of these memory devices to be mastered.

To start with we'll look at a typical RAM IC, the 2102. The logic symbol for the device is shown in **Fig 1**. It has 10 address lines, which means that it can address 1024 memory locations ($2^{10} = 1024$). It only has one Data input line however, which implies that each memory location can only store one bit. It is therefore referred to as being one bit wide and is organised as 1024 x 1 bits. In common with nearly all memories it has two control lines; a chip enable or chip select line (CE) and a read/write line (R/W). The final line shown in **Fig 1** is the Data output line.

In order to write a single bit of information into the memory, the appropriate address is set up on the 10 address lines while the CE and R/W lines are both HIGH (i.e. the chip is NOT yet enabled). The data is then set up on the data input line and the CE and R/W are both taken LOW for about 100ns during which time the data is written into the appropriate register in the memory. Reading from the memory is equally straightforward. The desired address is again placed on the 10 address lines but this time the read/write line is left HIGH, and only the chip enable line is taken LOW. After a maximum of 100ns the data which was previously read into the location will appear on the data output line. Incidentally, while both the control lines are HIGH, the data output line (which is a tri-state output) is in its high-impedance state.

How it works

The 2102s logic diagram is shown in **Fig 2**. Each memory location is a D-type latch which can store one bit. The 10 address lines are decoded into 1024 separate lines which are used to gate both the clock inputs to each of the latches, and the outputs from the latches. (In practice, the latches are arranged on the chip in a rectangular array of 32 rows and 32 columns, with 5 address lines connected to each of two one-of-32 decoders. Since there are, however, 1024 unique crossing points, we may think of this arrangement as being equivalent to a single one-of-1024 decoder). The clock pulse which causes a bit to be read into one of the latches is derived from a gate which goes HIGH only when both the read/write and the chip enable line are LOW (i.e. a NOR gate). Because the clock pulse must reach only one of the latches, this output is ANDed with one of the outputs from the one-of-1024 decoder (which, for the purposes of this description, are assumed to be active HIGH). When this pulse reaches the appropriate latch, the current state of the data input line is written into the latch. As soon as either the read/write line or the chip enable line goes HIGH again, the clock lines all go LOW and the latches are all locked out preventing any data from being written until the next time.

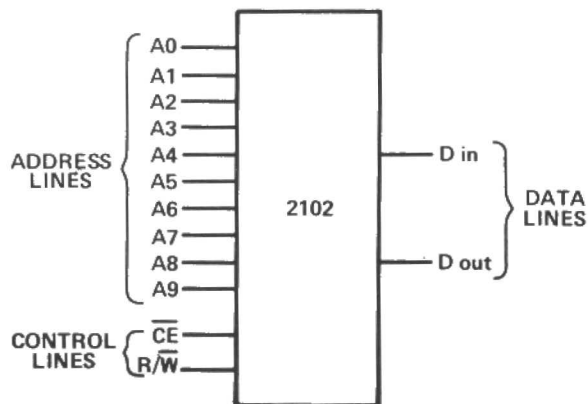


Figure 1. The logic system for a typical RAM IC – the 2102.

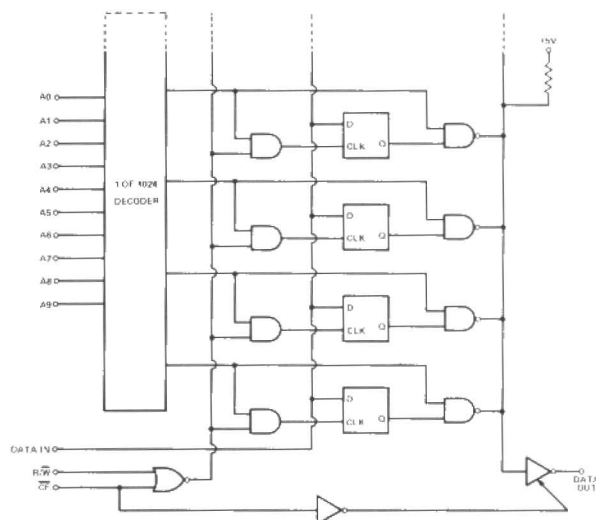
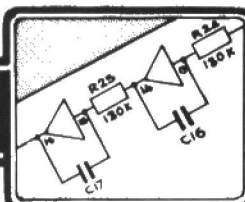


Figure 2. The logic diagram for the 2102 IC.

When a particular latch needs to be read, the appropriate address is again placed on the address lines causing one of the NAND gates on the outputs of the latches to become enabled. This output is the one required but, because all the other NAND gates are disabled, all their outputs will be HIGH. Somehow, all these outputs must be ANDed together so that the HIGH outputs will not swamp the one we need. Now a 1024-input AND gate is quite impracticable, and in any case there is a much easier way. The NAND gates which gate the



outputs from the latches are a special type known as open-collector outputs. They have the property that they do not actively assert a HIGH output, but they do pull the output LOW when required. Any number of these outputs may be connected together and if any one or more outputs go LOW, the overall output will be LOW. The only slight trouble is that when all the outputs are HIGH, the resultant output will float rather than properly asserting a HIGH state. For this reason, a resistor is connected from the common output line to the positive supply rail. This arrangement is sometimes called 'wired-AND' because one gains the effect of an AND gate simply by wiring the previous outputs together.

Finally, the now single data output line is passed through an inverting tri-state buffer. An inverting buffer is used to cancel out the effect of the previous NAND gate. This buffer is enabled only when the chip enable line is LOW – hence the need for the inverter in its own enable line. It is worth mentioning that this buffer is enabled during write as well as read operations. Data can therefore be read and written (to a particular latch) simultaneously.

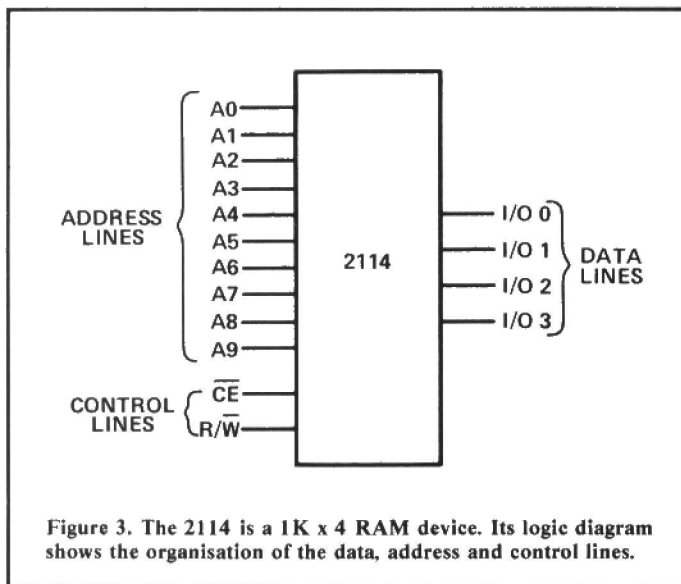


Figure 3. The 2114 is a 1K x 4 RAM device. Its logic diagram shows the organisation of the data, address and control lines.

Greater capacity

The 2102 is not a particularly suitable chip to use in a microcomputer because it is only one bit wide. In order to store 1K bytes of information, it would be necessary to have eight chips, each handling one of the eight bits in a single byte. The 2114, which is used in a number of home computers stores four times as many bits as the 2102 and yet it needs only two more pins. How is it done? The logic symbol shown in Fig 3 shows the basic principle. Like the 2102, the device has 10 address lines, but instead of having one data input and one data output, it has four data lines which serve as both input and output. This arrangement which uses the same pins for both purposes is ideal for a computer which has a single data bus carrying data to and from the microprocessor. The outputs are, of course, tri-state.

The 2114s logic diagram is shown in Fig 4, and is only a little more complex than that of the 2102. Instead of just one

latch, each decoded address line goes to four latches. The clock line (for writing) is decoded in exactly the same way as before and is connected to all four latches in one register so that all are written to simultaneously. The outputs from the latches are Nanded as before and their outputs commoned together in the familiar wired-AND arrangement. The only real difference is that two sets of tri-state buffers are provided, one on the input lines and one on the output lines. These are gated in such a way that both are disabled when the chip enable line is HIGH, and when the chip enable line is LOW, only one set is enabled depending on the state of the read/write line. For the purposes of the diagram, the input buffers are shown as non-inverting types, while the output buffers are inverting. This is to ensure that the data which is read is of the same polarity as the data written. (It should be borne in mind that a diagram like this is merely illustrative and should not be regarded as a necessarily exact representation of what is to be found inside a real chip. In any case, different manufacturers are at liberty to organise the insides of their chips how they like; what is important is an understanding of their workings from an operation point of view).

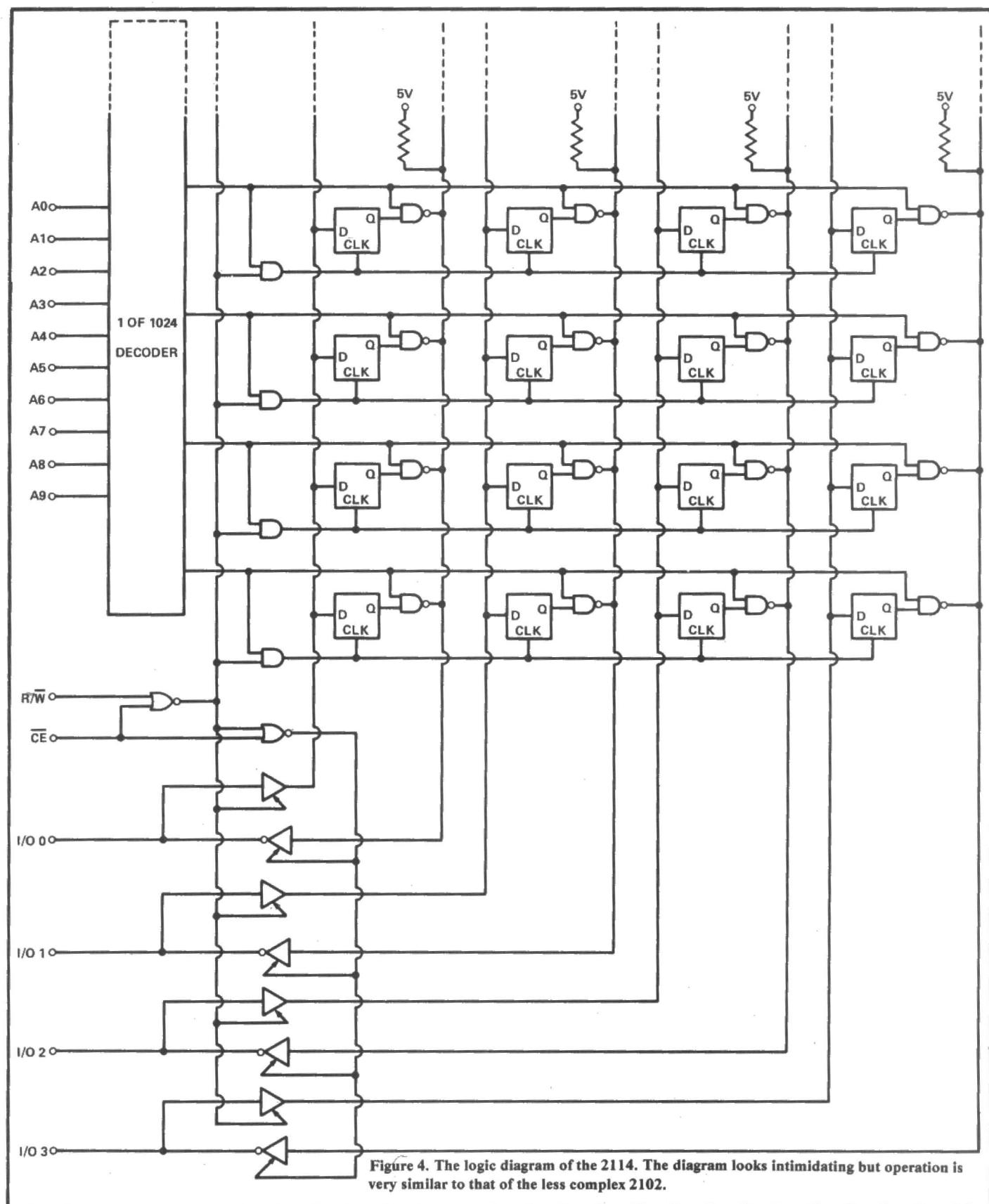
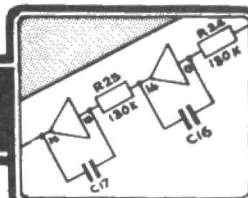
Still more memory

Suppose now that we wish to use a number of 2114s to make a 4K RAM – i.e. 4096 words, each word being a single byte of eight bits. We will require an eight bit data bus (D0-D7) and a twelve bit address bus (A0-A11, $2^{12} = 4096$). Since the 2114 has only four data lines, we must use them in pairs, each device only storing half a complete word. As for the address lines, A0 to A9 can be connected to all the chips at once, but the other two lines A10 and A11 must be decoded so as to enable only one pair of chips at any one time. The read/write lines can also be connected together. The complete circuit is shown in Fig 5.

We have seen earlier in this series that a D type latch requires six or more gates. It follows that a single 2114 contains about 24,000 transistors, not including the necessary decoding circuitry. If it was built using standard TTL integrated circuit technology it would be far too large. At present, memories of this size are made under large scale integration (LSI) using NMOS gates. Readers may be familiar with the other major logic family – CMOS or Complementary Metal Oxide Silicon logic. NMOS is a version of this family which uses n-type MOS transistors only, the advantage being that the transistors can be made much smaller. A lot more data could be stored on a chip if it was possible to use just one transistor to store each bit. This can be done using NMOS Field Effect Transistors in a way that does not use D-type latches at all.

Dynamic Devices

Field Effect Transistors (FETs) have an exceedingly high input impedance – that is to say, the base of such a transistor takes a very small current indeed. The base can also be regarded as a very small capacitor which can store a minute quantity of charge that can keep the transistor on. A single FET can therefore act as a memory element storing a 1 when the capacitor is charged and the transistor ON, and a 0 when the capacitor is discharged and the transistor OFF. The trouble is that, in spite of the very high impedance of the



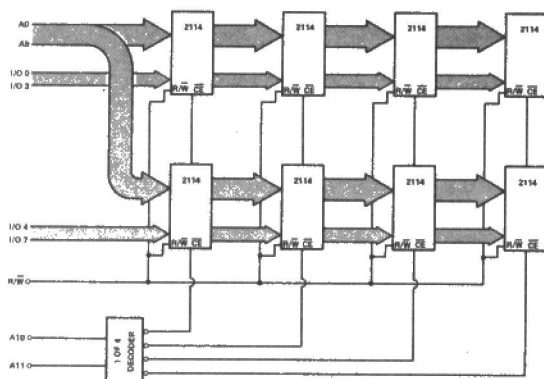
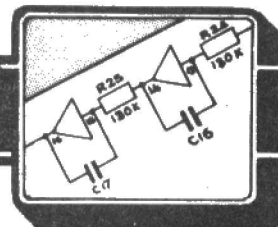


Figure 5. By using a number of 2114 devices, a practical, byte wide, memory system can be realised.

transistor, the charge still manages to leak away quite quickly. It can, however, be replaced if the memory is refreshed every 2ms or so by accessing each of the elements in turn. Such a memory is called dynamic rather than static. Its advantage is that you can get four times as much memory on the same area of silicon and at the same price as with static RAM but the associated circuitry is rather more complex owing to the necessity to provide the refresh signals. The 4116, for example, stores 16,384 single bit words and just eight of these devices would add 16K of memory to your computer. The 64K-bit memory already exists and larger memories are well on the way. The 4864 is a 64K-bit memory that combines the advantages of both static and dynamic memory. It is in fact a dynamic memory with single transistor storage elements, but it contains on chip most of the extra circuitry necessary to refresh the memory so that it appears to behave like a static memory. Because of their high storage density and ease of use we can expect to see a lot more of these devices in the near future.

EPROMS Explained

An EPROM like the popular 2716 is very similar to a dynamic memory in that it too uses n-channel MOSFET transistors as storage elements. The write circuitry is, however, simplified so that the charge on the base leaks away very slowly indeed – with a time constant measured in tens of years rather than milli-seconds. Once programmed, therefore, the device will retain its data even when the power is turned off. A logic symbol for the 2716 is shown in **Fig 6**.

In normal (read only) mode, V_{pp} is held at 5V and \overline{CE}/PGM is LOW (i.e. the chip is enabled for reading, not writing). When the OE pin is HIGH, the tri-state outputs are in their high-impedance state so that they do not interfere with any other data that might be on the data bus. When it is necessary to read a byte of data from the memory, the appropriate address is set up on the address lines and the OE (output enable) pin is taken LOW. The data then appears on the output lines. In this respect, the 2716 behaves like an ordinary memory with the output enable pin doing the job of the more conventional chip enable pin.

Programming the 2716 is a different matter. First, a 25V supply is connected to the pin named V_{pp} . \overline{CE} must be LOW

and OE must be HIGH. Under these circumstances, the output lines double as input lines as well. When the appropriate address and data lines have been set up, \overline{CE}/PGM is taken HIGH for between 45ms and 55ms. This applies a 25V pulse to the appropriate transistors, charging up their base capacitors by a process known as 'avalanche breakdown'.

It does not concern the user very much but in fact a 1 is represented by a **discharged** capacitor and it is the 0s which are read into the memory. This explains why a blank EPROM appears to be full of 255s rather than zeroes. The data can be verified even while V_{pp} is at 25V simply by taking OE LOW again. These different modes are summarised in **Table 1**.

Erasing an EPROM is accomplished by placing it under a strong ultra-violet light for about 20 minutes. The UV light literally knocks the electrons out of the base capacitors thus effectively discharging them. It should be remembered that sunlight and even fluorescent light may contain sufficient UV light to cause partial erasure.

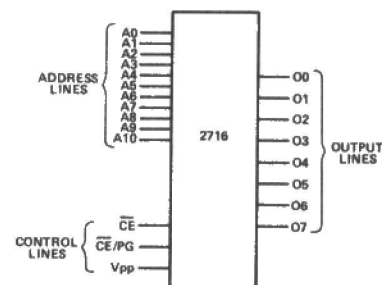


Figure 6. The lock symbol of the 2716 EPROM that will retain its data when the power is turned off.

Table 1

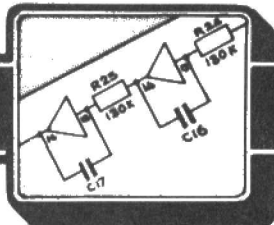
V_{pp}	\overline{CE}/PGM	OE	Mode	Outputs
5v	HIGH	HIGH	Standby	High impedance
5v	HIGH	LOW	Standby	High impedance
5v	LOW	HIGH	Deselect	High impedance
5v	LOW	LOW	Read	Data out
25v	HIGH	HIGH	Program	Data in
25v	HIGH	LOW	Standby	High impedance
25v	LOW	HIGH	Inhibit	High impedance
25v	LOW	LOW	Read	Data out

2716 EPROM modes

To the future

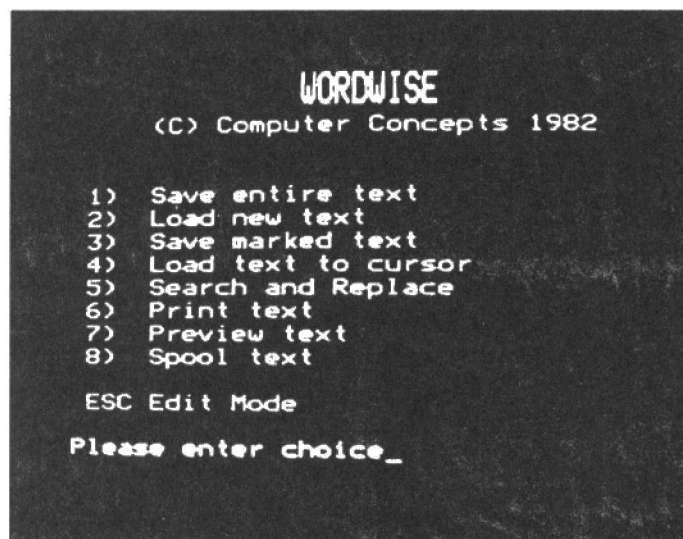
As with all memories, larger and larger versions are on their way and the cost per bit keeps tumbling. The day is perhaps not far off when the standard method of storing programs for the home micro will be on plug-in modules containing a 16K dynamic memory chip with its own battery power supply to retain the data when not in use.

E&CM



Wordwise In Action

Of micros in 'serious' applications, about thirty per cent are used for Word Processing. S. M. Gee reviews one of the most popular WP packages for the BBC micro – Computer Concept's Wordwise.



Wordwise has been on the market for some months and has attracted a lot of interest by being the first word processing package available for the BBC Micro – it has also attracted imitators who, it seems, have copied it, bugs and all! (Not that I have found any bugs that affect its operation whatsoever).

Wordwise is extremely easy to begin to use and even a complete newcomer to word processing should find very few problems in getting started. Once WORDWISE is installed you can get into it from BASIC by typing ★WORDWISE. The first message displayed on the screen is "Old text? y/n". At the beginning of a word processing session the answer to this has to be "n". (You would only answer "y" if you had broken off in the middle of working on some text and it was still in the BBC Micro's memory). The user is then shown the menu which offers eight "menu" options (which we will return to later) and the "edit" mode. In order to enter text into the word processor, you should select the edit mode, by pressing the ESCAPE key.

When in the edit mode, the text that you type in appears in white in teletext graphics (Mode 7) which gives a clear display on the screen but has the disadvantage of having a maximum of 40 characters per line. This is a problem that you can disregard entirely, however, as the final layout of your document on paper is entirely under your control and you can "preview" the way it will look at any time simply by pressing two keys.

When using Wordwise there is a clear distinction between formatting commands and editing facilities. This is one feature that makes it particularly user-friendly for "normal" applications, such as writing text without many headings, indents and tabulations. The editing facilities are to do with the text that appears on the screen in front of you and the formatting commands govern how the end product will look.

As mentioned earlier, the first advantage that a word processor has over a typewriter is that you no longer have to worry about silly slips of the finger as you type. This means in general that you can type very much more quickly than on a conventional typewriter.

By using the cursor control keys you can place the cursor anywhere on the screen. Wordwise combines use of the SHIFT and CTRL keys and the cursor keys in a simple and systematic way which enables you to scan through the text at speed in any direction.

Correcting Mistakes

To return to the subject of correcting your typing errors, you can insert new characters directly above the cursor or, if you have chosen the "over" option, you can overwrite the existing character. To delete characters you can either use the DELETE key, which deletes the character to the immediate left of the cursor, or you can press CTRL and A simultaneously to delete the character directly above the cursor. There is also a facility for deleting the whole word above the cursor. The remaining editing command is one I found particularly welcome, having been used to another word processor with no similar facility – it is the ability to change the case of the character immediately above the cursor by pressing CTRL and S.

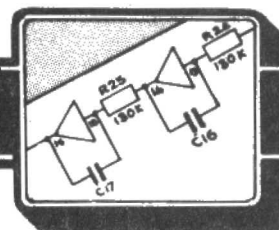
In addition to the methods discussed above, there are two ways of deleting larger chunks of text. Both of these rely on using the BBC Micro's function keys, all of which have a pre-defined use in Wordwise. Until you actually have a chance to try out WORDWISE for yourself, the use it makes of the function keys is probably difficult to appreciate. However, this is one of the features that makes it so easy to use. In the case of WORDWISE each function key has a single meaning which is accessed by pressing the function key on its own. A prompt card is provided to be inserted under the clear plastic strip just above the function keys on the BBC Micro. You line this up with the keys so that each one is properly labelled.

In WORDWISE the function keys have the meanings indicated in Table 1.

Table 1

f0 INSERT or OVER	f5 WORD COUNT TO?
f1 GREEN	f6 DELETE TO?
f2 WHITE	f7 DELETE MARKED TEXT
f3 MARKER	f8 MOVE MARKED TEXT
f4 CURSOR TO?	f9 COPY MARKED TEXT

Pressing f0 has the effect of switching from insert mode to the overwriting mode and vice versa. When you first enter Wordwise you are automatically in insert mode and each time you press f0 a bleep alerts you to the fact that you have switched modes. The two keys f1 and f2 are used when you



wish to use "embedded" commands – more of which shortly. The markers that are inserted when you press f3 are used to enclose passages that you wish to delete or to move or copy, using f7, f8 and f9 respectively. These facilities give you the freedom to restructure text once you've typed it onto the screen or to repeat passages that recur within the same document. The remaining three function keys all search through the text to find a specific character, in order to move the cursor to that point (f4), to count the words between two points (f5), or to delete all the text up to that point (f6). This search facility only recognises a single character which means that its utility is limited. In fact the only viable way to use it is to insert a character that you otherwise would not use at the point you are interested in prior to using one of these keys. If, alternatively, you could supply a string when asked to indicate the target point, the facility might be more generally useful. If you attempt to delete more than 250 characters a warning note is sounded and the question "Are you sure?" appears. This is a comforting feature as it is very frustrating to delete your text when your finger slipped when you really wanted a word count!

Formatting Text

So far, I have considered the editing facilities available on Wordwise and found them to be easy to get used to. Now it is time to consider its ability to format text on the printed page. This is where the "embedded" commands come into their own. It is the embedded commands that make using a word processor different from using a typewriter, so it is at this point that readers who have never used a word processor may begin to find my explanation unfamiliar.

While you are typing in your text any embedded commands have no effect on the layout of your words on the screen. Instead they appear in green exactly as you typed them. Only when the document is printed, previewed or spooled (stored on disc or cassette in a form ready for printing) are these two letter commands translated to format your words as you determined. Until you get used to this idea it can be quite confusing and this is one reason that I like the easy way in which you can preview your output with Wordwise.

To a large extent, it is the range of embedded commands that determine how useful a word processor is and WORDWISE has a sensible set of them. Moreover, the values chosen as the "default values", which WORDWISE assumes unless you specify otherwise, mean that you often do not need to worry about selecting options. So for example, Wordwise starts out with justification on – that is with both right and left margins set, with a page length of 66 lines (just right for standard computer stationery) and with a line length of 70 characters (again, just right for the majority of printers that are suitable for word processing). Personally I would be happier with the command that causes the processor to split text up into pages to be ON rather than OFF unless specified otherwise but I imagine others prefer it as it is.

I was very pleased that the two letter abbreviations chosen for the embedded commands make sense and are therefore easy to remember. For example, you type CE to centre lines, LM for left margin, IN for indent. (I may, of course be biased because these commands have exactly the same mnemonics as the word processor I am already familiar with!)

You may remember mention at the beginning of this section of the eight "menu" options available in Wordwise.

These allow you to do something with your text once you've typed it all in. The menu appears on the screen as shown and the options are fairly self-explanatory. The first two options enable you to save and load text, the third one enables you to save just part of the file currently in memory and the fourth one allows you to load text within the file currently held in memory – thus providing a way of merging files. Option 5 provides a facility that will be familiar to users of other context editors. It allows you to search through the file for occurrences of specific strings and to replace either all or some of them by an alternative string. This can be very useful, for example, for correcting systematic misspellings or to rewrite letters to be sent to other recipients. Option 6 sends the text to a printer, option 7, which I find so very useful, sends exactly the same output to an eighty column screen (except that when the text becomes so large that there is insufficient memory left for this it gets restricted to mode 7) and option 8 sends it to a filing system – disc or cassette. There is one other very important facility available from the menu – you can gain access to all the BBC Micro's operating and filing system commands by typing ★ followed by the command. So if you want to catalogue your disc prior to load or saving you can simply type ★CAT, while ★BASIC returns you to BASIC.

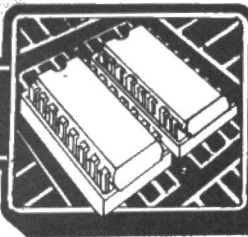
One word of warning when using the menu options, do not type 2 (load) when you actually mean 1 (save) AND supply a filename. If you do you will overwrite your text in memory with absolutely nothing – I discovered this when it happened to me! Had I noticed my mistake before I supplied a filename, pressing BREAK and answering "Y" to the question "Old text" would have averted disaster but no provision is made for out and out stupidity!

Wordwise uses colour to good effect – and not only if you have a colour TV as the colours it uses will also stand out on a black and white set. As already explained, green is used for the embedded commands. The beginning and end of the text you are working on are indicated by the words "start" and "end" displayed in red and the "status line" at the top of the screen is blue. This informs the user how many words have been written and how many characters remain free. This is important since you can run out of memory if you write a lengthy document. At the far left of the status line a yellow "I" is normally displayed to indicate that you are inserting text. If you press f0 to overwrite existing text this changes to "O". Wordwise also uses sound to alert you to situations you probably do not intend.

This review would not be complete unless I mentioned Wordwise's main limitation. One application that many word processors are used for is to despatch "form letters", that is letters that are nearly identical in content but differ in whom they are addressed to and in other points of content.

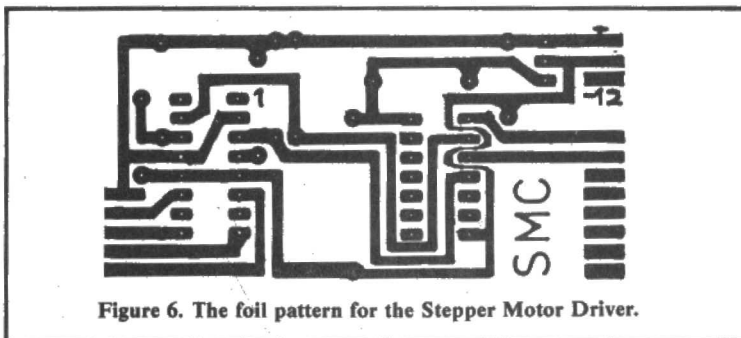
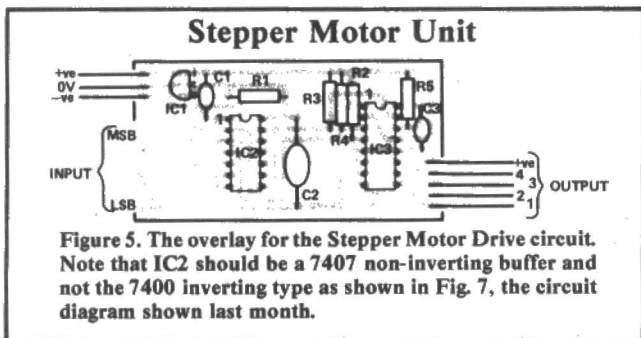
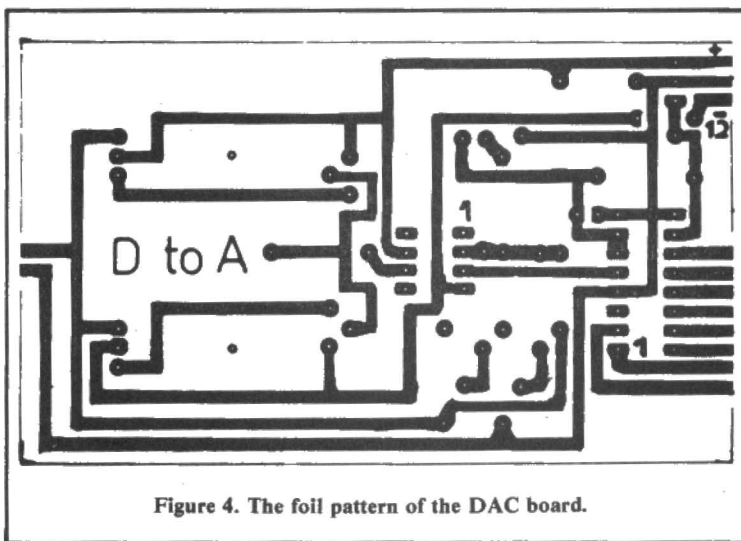
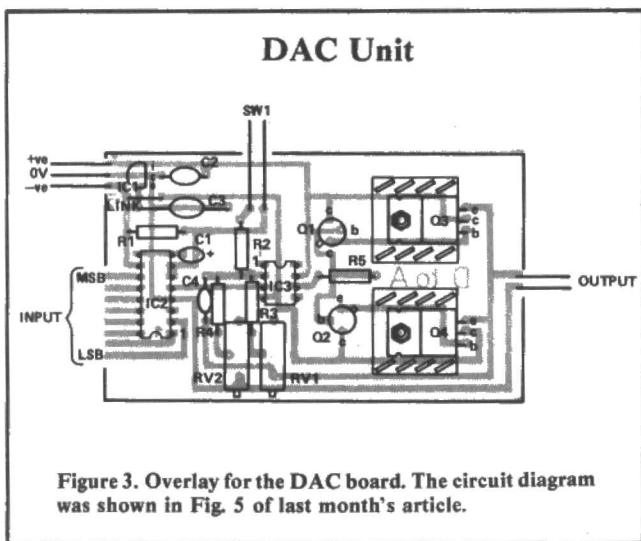
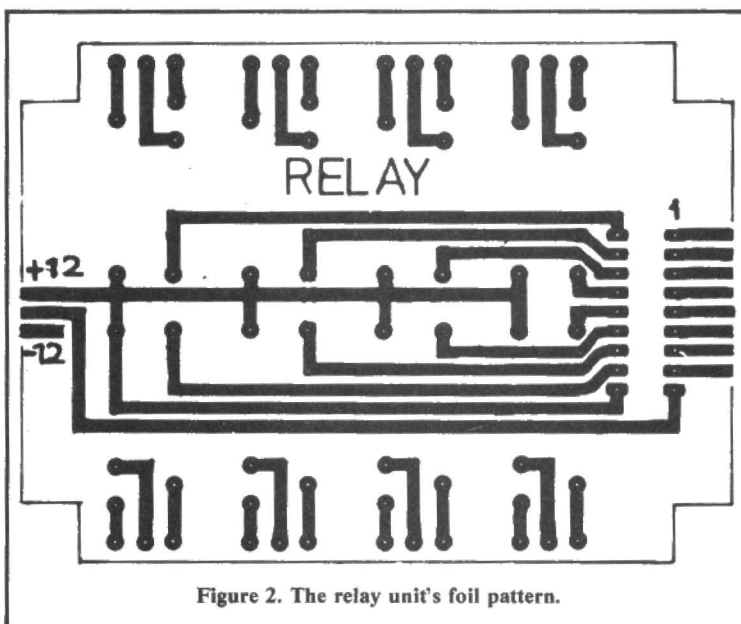
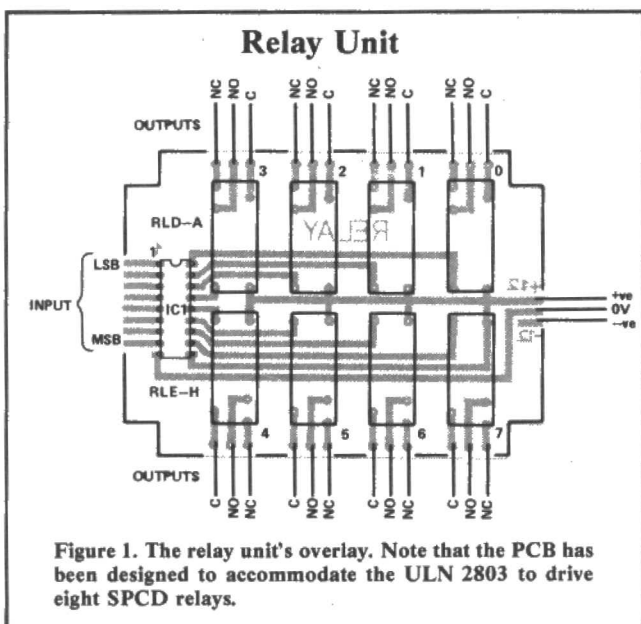
A comparison of the two packages will be presented after View has been examined next month. In finishing this month, I'll just say that Wordwise offers an extremely user friendly program that is sufficiently powerful for most everyday applications. Its lack of a way in handling large files in a straightforward way is a disadvantage but this must be weighed against its main attraction – it is very easy to use. It would make an excellent choice for someone rather apprehensive about word processing.

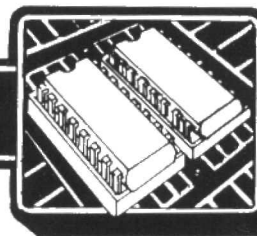
E&CM



POWER CONTROL FOR MICROS

Last month Malcolm Plant described three designs that enable micros to interface with power handling devices. This month, as promised, we present PCB layouts for each of the designs in order to simplify their construction.





Spectrum Printer Interface

Our interface allows printers with a parallel interface to be driven by the Spectrum. Design by R. Sargent.

When Sinclair's RS232 interface finally sees the light of day, it will be possible to obtain quality printout from the computer by means of a printer with an RS232 interface. Printers with such an interface are, however, some £30 to £40 more expensive than equivalent machines with a parallel interface. With the Sinclair adaptor itself due to sell for £25, it would seem worthwhile to consider the parallel approach.

Design outline

The interface board uses the Z80 Parallel Input and Output chip which is an extremely versatile device. Construction is straightforward using this particular IC and many software configurations are possible; it is well worth obtaining Zilog's technical manual for a full run down on what is available. The main purpose of this article though is to get the Spectrum running a parallel printer interface; the code that runs the printer is only a small part of the program.

A parallel port can be added very easily to the Spectrum and will occupy four consecutive locations in the Input/Output Port Map. The Z80 PIO generates its own "data valid" or STROBE signal and allows the computer to output all 8 bits as data. Link 1 to 5 and 2 to 3 on the Strobe select link pins, and alter the software as instructed in the graphics section later in the text. An alternative approach is to output 7 bits of data, the eighth bit being used as the strobe. In this case the length of the strobe is adjustable using software. Link 1 to 3 and 4 to 5 on the Strobe select link pins. One input line is needed to monitor the printer's BUSY signal; the other 7 inputs lines are free for other applications. **Fig 2** shows the circuit for the interface. The prototype was constructed using a wiring pen on a piece of stripboard three inches square. Layout is not critical and decoupling capacitors are unnecessary.

Address	A7	A6	A5	A4	A3	A2	A1	A0	Function
xx1F	0	0	0	1	1	1	1	1	Available to the user
xx3F	0	0	1	1	1	1	1	1	
xx5F	0	1	0	1	1	1	1	1	
xx7F	0	1	1	1	1	1	1	1	
xx9F	1	0	0	1	1	1	1	1	
xxBF	1	0	1	1	1	1	1	1	Read & Write to S.P. ★ note 1
xxDF	1	1	0	1	1	1	1	1	
00EF (239)	1	1	1	0	1	1	1	1	
00F7 (247)	1	1	1	1	0	1	1	1	
00FB (251)†	1	1	1	1	1	0	1	1	
00FD (253)	1	1	1	1	1	1	0	1	Read & Write to S.P. ★ note 1
00FE (254)	1	1	1	1	1	1	1	0	Write to Beeper/Ear Socket
xxFE	1	1	1	1	1	1	1	0	Read the Keyboard. ★ note 2
xxFF	1	1	1	1	1	1	1	1	Available to the user

Note 1: S.P. = one of the Spectrum Peripherals
Network System
RS232 Interface
Microdrive

Note 2: When a READ is made to the low address byte FE, the BITS of the high address byte are taken low in sequence to strobe the keyboard. The addresses involved are: 7FFE (32766), BFFE (49150), DFFE (57342), EFFE (61438), F7FE (63486), FBFE (64510), FEFE (65278).

Table 1. Partial decoding of the port addresses gives a choice location for the printer.

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
x	x	x	x	x	x	0	0	0	0	0	1	1	1	1	1	001FH (31) Port 1
x	x	x	x	x	x	0	1	0	0	0	1	1	1	1	1	011FH (287) Port 2
x	x	x	x	x	x	1	0	0	0	0	1	1	1	1	1	021FH (543) Port 3
x	x	x	x	x	x	1	1	0	0	0	1	1	1	1	1	031FH (799) Port 4

x = don't care
take it as 0

Table 2. A guide to the addresses of the four ports.

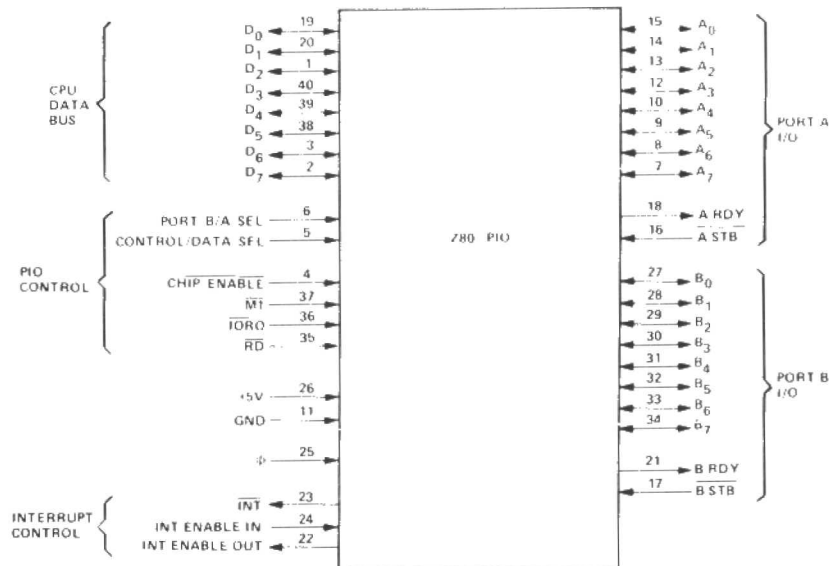
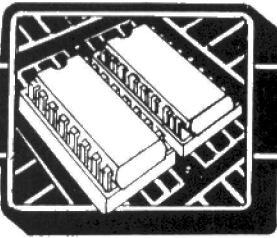


FIGURE 3.0-1
PIO PIN CONFIGURATION

Figure 1. The pin-out for the Z80 PIO.



Untangling the ports

The Spectrum makes good use of its I/O ports, but there is still plenty of address space available for the user. However, there is no way of telling whether the I/O address chosen for a do-it-yourself add-on will not clash with the I/O address chosen for any other add-on . . . The answer lies in the flexibility offered by jumper wires, so that the PIO port address for the printer can be changed if necessary.

Table 1 shows how even the partial decoding of the port addresses gives a good choice of locations for the printer interface. It is clear that if one or more of lines A5, A6 and A7 are kept low then there will be no conflict with the Spectrum peripherals.

Partial decoding

Address lines A5, A6 and A7 are taken to IC1, a 3 to 8 decoder and give 8 imperfectly decoded outputs. Seven of these outputs would probably be suitable to use, but the eighth is certainly not, as it is enabled when A5, A6 and A7 are in the process of accessing a Spectrum peripheral. A safeguard which overcomes this problem is provided by gates IC2a, IC2b and IC3a which provide a negative-logic AND function on lines A0, A1, A2, A3, A4 and IORQ, and so enable IC1 at the correct time. Each output from IC1 therefore has a single address, unique within the first 256 bytes of the port map, but repeated in every 256-block throughout the entire map.

The Z80 PIO chip (IC4) contains 4 separate ports which must be addressed,

and any of lines A8 to A15 are suitable. Lines A0 to A7 are responsible for enabling the chip (via IC1). A table such as **Table 2** is invaluable in settling the final addresses of the four ports.

The software

The routine to send a Spectrum BASIC listing to a parallel printer is made rather complicated by the fact there is no printer "reflection" in the Spectrum workspace. A reflection is a value held in a double or triple byte of RAM through which the program normally flowing in ROM has to pass; alter the value in the RAM bytes, and the user can divert a monitor routine and use it to his own ends. Since we cannot break into the Spectrum ROM, the printer program has to do everything itself, which takes about a half K of memory. What is everything? It involves changing the line

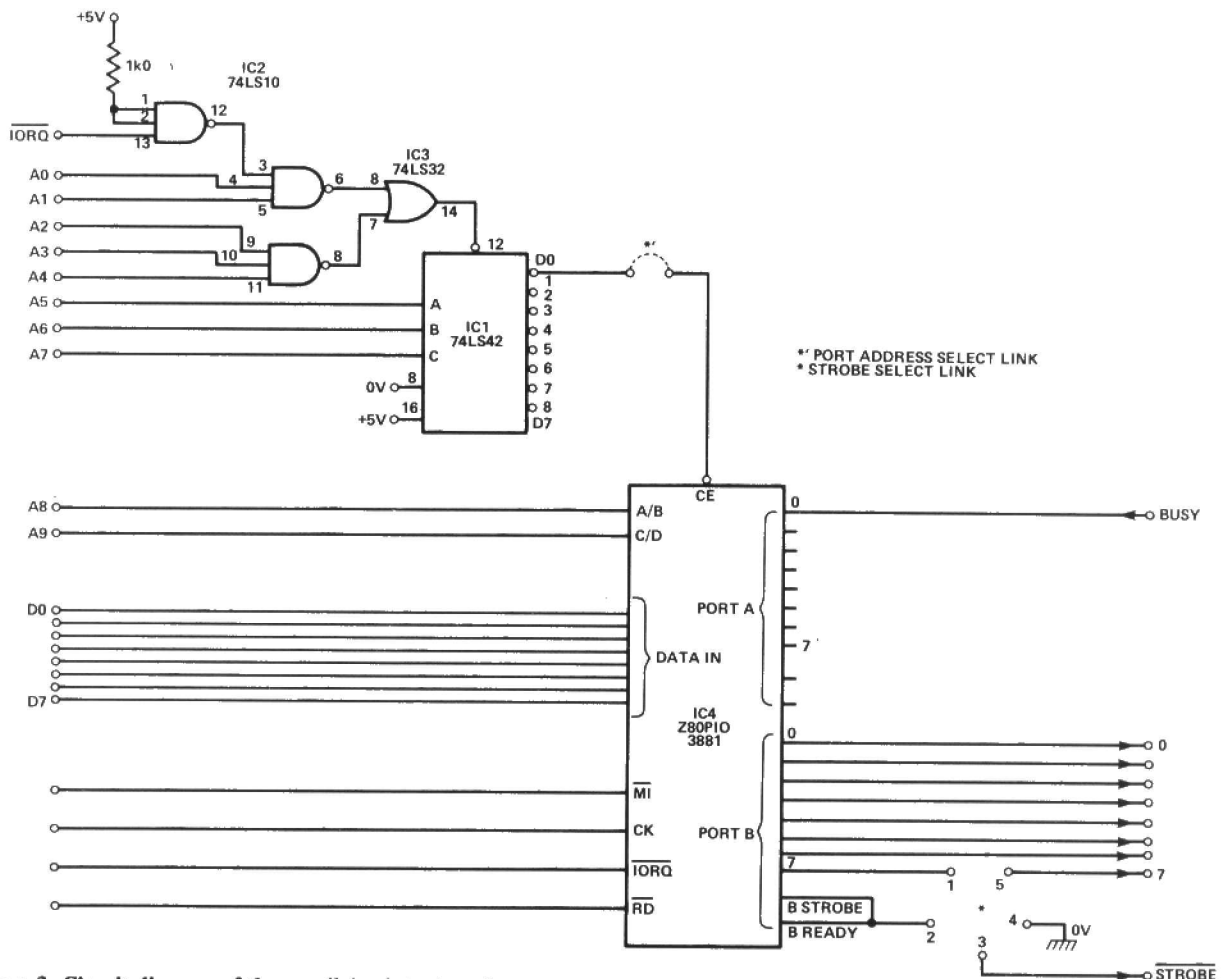
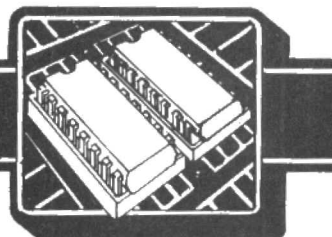


Figure 2. Circuit diagram of the parallel printer interface.



numbers from their binary representation to decimal; single-byte tokens have to be decoded and expanded into their full form (e.g. **RANDOMIZE**); quite a few bytes must not be sent to the printer at all; some, like the block graphics, have to be changed before they go to the printer . . . and so on. A study of the source listing will show how this is done.

A short BASIC program loads the printer routine, which is in machine code. The same program resets HIMEM so that the machine code program is safe, and SAVES the CODE to a cassette for future use. If you have access to a machine code loader, then the HEX CODE may be typed in directly, and this will take far less time than typing in the decimal code within the BASIC data statements. However, owners of 48K Spectrums will have to do a spot of work if they want to locate

the program at the top end of their memory. There are 12 absolute jumps, 29 calls and 2 bytes of code at 32234 which will all need adjusting. Their new position will be 32768 bytes further on in memory.

Using the software

The Store and USR values given are for the 16K Spectrum. Add 32768 to each to get the 48K Spectrum values, with the exception of the two print buffer values which stay the same.

Emulating LLIST

RANDOMIZE USR 32050 will print out a full listing of a Basic program. This is an immediate mode command and randomize is used because the USR function always returns to Basic with the value held in its BC register, and that value expects to be put somewhere!

Emulating LPRINT

The Basic program must send the characters it wishes to LPRINT to the print buffer by a series of POKES. The print buffer starts at 23296 and finishes 256 bytes later at 23552. When sufficient characters have been sent there, the carriage return character (13) is also sent and the program does a LET A =USR 32053, which prints the contents of the buffer up to and including the carriage return.

PRINT A TEXT FILE/ HEXADECIMAL DUMP

The starting byte of the file is **POKED** into 32072, and the byte after the last carriage return is **POKED** into 32074. With store 32069 set at zero.

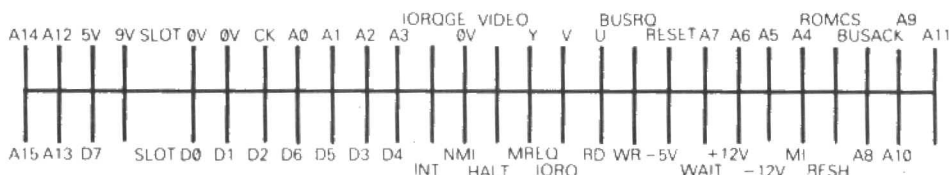


Figure 3. Connection details of the Spectrum's expansion connector.

The complete program listing

OPTIONS=OPTIONS=7131

```

7131                                     The complete program listing
7131
7131
7131
7131                                     ;*****
7131                                     ;#      SPECTRUM TO EPSON Feb-83      #
7131                                     ;#              R GARGENT          #
7131                                     ;***** V3 *****
7131
7131
7F57   RANTOP    EQU 32599             ;16K spectrum 7F57H/32599
7F57                                           ;48K spectrum FF57H/65367
7F57
7D32   USR       EQU RANTOP-225H
5C53   PR0G      EQU 23635            ;start information, basic file
5C53   VAR0      EQU 23627            ;end information, basic file
0096   TAB1      EQU 150              ;reserved word lookup table
5B00   BUFF      EQU 32296           ;printer buffer
5BFf   BUFFEND   EQU 23551           ;last byte of printer buffer
002E   SYM       EQU 2EH              ;default symbol for
002E                                           ;unprintable characters
002E
7D32   DRG       EQU 0
7D32   LOAD USR
7D32
7D32   BEGIN     EQU $ ;first some various starts
7D32
7D32 C34F7D      JP PLIST ;print out basic file
7D35 C5827E      JP PBUF  ;print out the (preloaded) buffer
7D38 C3D77E      JP PHEN  ;print out section of memory
7D3B C3557E      JP PBYTE ;print a single byte
7D3E C3557E      JP PBYTE ; spare
7D41 C3557E      JP PBYTE
7D41
7D41                                     ;WORKSPACE (may be altered)
7D41
7D44 50          MAXLEN DB 00 ;characters per line
7D45 00          HEXD   DB 0  ;0 print byte
7D45                                     ;1 print byte hex notation
7D46 00          ASTORE DB 0  ;character to be printed
7D47 00          SWI    DB 0  ;0 no LF after CR
7D47                                     ;1 one LF
7D47                                     ;2 two LFs
7D48 0000        START DW 0
7D4A 0000        FINISH DW 0
7D4A
7D4A                                     ;SCRATCH WORKSPACE
7D4A
7D4C 00          PRHEAD DB 0 ;print head position
7D4D 0000        POSIT  DW 0 ;position within file
7D4D
7D4D

```

```

7D40
7D4F 2A535C PLIST LD HL,(PROG)
7D52 22487D LD (START),HL
7D55 2A485C LD HL,(VARS)
7D58 224A7D LD (FINISH),HL
7D5B CDC17E PLA CALL INIT
7D5D
7D5E CCCC7E BASCAN CALL TEST
7D61 CB RET Z;if (FINISH)>(START).return to Basic
7D61
7D61 ;Convert line number
7D62 1A LD A,(DE)
7D63 67 LD H,A
7D64 13 INC DE
7D65 1A LD A,(DE)
7D66 6F LD L,A ;hex line num now in HL
7D67 113838 LD DE,3838H ;load 8888 as ascii
7D6A 813838 LD BC,3838H
7D6D 7C LD D,H
7D6E B5 CDOWN OR L
7D6F 281F JR Z L0 ;---> finish conversion when HL=0
7D71 28 DEC HL ;countdown the hex
7D72 8C INC C ;countup the decimal ascii
7D73 79 LD A,C
7D74 FE3A CP 3AH
7D76 28F5 JR NZ CDOWN
7D78 3E38 LD A,38H ;reset units to 8
7D7A 4F LD C,A
7D7B 84 INC B ;and increment tens
7D7C 78 LD A,B
7D7D FE3A CP 3AH
7D7F 28EC JR NZ CDOWN
7D81 3E38 LD A,38H ;reset tens to 8
7D83 47 LD B,A
7D84 1C INC E ;and increment the hundreds
7D85 78 LD A,E
7D86 FE3A CP 3AH
7D88 28E3 JR NZ CDOWN
7D8A 3E38 LD A,38H ;reset hundreds to 8
7D8C 5F LD E,A
7D8D 14 INC D ;and increment the thousands
7D8E 18DD JR CDOWN
7D8E
7D8E ;Output line number,
7D8E ;suppressing leading zero
7D90 7A L0 LD A,D
7D91 FE38 CP 38H
7D93 2882 JR NZ L1
7D95 3E28 LD A,28H
7D97 CD587E L1 CALL SEND
7D9A 7B LD A,E

```

```

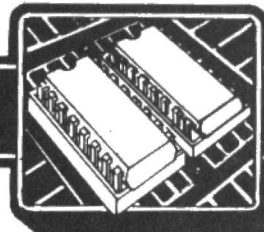
7D9B CD587E      CALL SEND
7D9E 78          LD A,B
7D9F CD587E      CALL SEND
7DA2 79          LD A,C
7DA3 CD587E      CALL SEND
7DA3
7DA3             ;Advance past "length of text" bytes
7DA6 2A4D7D      LD HL,(POSIT)
7DA9 23          INC HL
7DAA 23          INC HL
7DAB 23          INC HL
7DAC 115E7D      LD DE,BASCAN ;return address onto stack
7DAF 05          PUSH DE
7DAF             ;fall through to Scan
7DAF
7DAF
7DAF
7DAF             ;Subroutine :
7DAF             ;Scan & print text until @DH reached
7DAF
7DB0 23          SCAN INC HL
7DB1 7E          LD A,(HL)
7DB2 FE0D        CP @DH
7DB4 2000        JR NZ C1
7DB6 CD587E      CALL SEND
7DB9 23          INC HL
7DBA 224D7D      LD (POSIT),HL
7DBD C9          RET ;-----> EXIT
7DBD
7DBD             ;check for the print comma
7DBE FE06        CP 6
7DBE 2007        JR NZ C2
7DC2 3E2C        LD A,","
7DC4 CD587E      CALL SEND
7DC7 18E7        JR SCAN
7DC7
7DC7             ;check for code 14
7DC9 FE0E        CP 14
7DCB 2007        JR NZ C3
7DCD 23          INC HL
7DCE 23          INC HL
7DCF 23          INC HL
7DD0 23          INC HL
7DD1 23          INC HL
7DD2 180C        JR SCAN
7DD2
7DD2             ;check for the control characters
7DD4 FE20        CP 31+1
7DD6 3002        JR NC,C4
7DD8 18D6        JR SCAN
7DD8
7DD8             ;check for block graphics
7DDA 181A        C4 JR C5 ;replace with @,@ if printer
7DDA             ;has 8 bit block graphics
7DDC @0          NOP
7DDD FE00        CP 128
7DDF 3015        JR C C5
7DE1 FE90        CP 143+1
7DE3 3011        JR NC C5
7DE3             ;output block graphics
7DE5 E5          PUSH HL
7DE6 1600        LD D,@
7DE8 5F          LD E,A
7DE9 21007E      LD HL,NW-128 ;address of newcode
7DEC B7          OR A
7DED 19          ADD HL,DE
7DEE 7E          LD A,(HL)
7DEF CD587E      CALL SEND
7DF2 E1          POP HL
7DF3 C3B07D      JP SCAN
7DF3
7DF3             ;check for more unprintable characters
7DF6 FE00        CP 128
7DF8 3000        JR C,C6
7DFA FE45        CP 164+1
7DFC 3007        JR NC,C6
7DFE 3E2E        LD A,BYM
7E00 CD587E      CALL SEND
7E03 10A0        JR SCAN
7E03
7E03             ;check for reserved word
7E05 FEFF        CP 255
7E07 2000        JR Z RESW
7E09 FE45        CP 163
7E0B 302E        JR C,C7
7E0D FEFF        CP 255
7E0F 302A        JR NC,C7
7E11 E5          PUSH HL
7E12 CD4D7E      CALL SPACE
7E15 219A00      LD HL,TAB1
7E18 D6A5        SUB 165
7E1A 2000        JR Z R3
7E1C 47          LD B,A
7E1D C07E        BIT 7,(HL)
7E1F 23          INC HL
7E20 20F0        JR Z R2
7E22 10F9        DNZ R2
7E24 7E          LD A,(HL)
7E25 C07F        BIT 7,A
7E27 2006        JR NZ R4
7E29 CD587E      CALL SEND
7E2C 23          INC HL
7E2D 10F5        JR R3
7E2F C00F        RES 7,A
7E31 CD587E      CALL SEND
7E34 CD4D7E      CALL SPACE
7E37 E1          POP HL
7E38 C3B07D      JP SCAN
7E38
7E38             ;check for copyright symbol
7E3B FE7F        CP 127
7E3D 2000        JR NZ OK
7E3F 3E63        LD A,"c"
7E41 CD587E      CALL SEND
7E44 C3B07D      JP SCAN
7E44
7E44             ;WHATEVER IS LEFT,,,
7E44             ;PRINT IT
7E44
7E47 CD587E      OK CALL SEND
7E4A C3B07D      JP SCAN
7E4A
7E4A
7E4A             SPACE PUSH AF
7E4E 3E20        LD A,20H
7E50 CD587E      CALL SEND
7E53 F1          POP AF
7E54 C9          RET
7E54

```

```

7E54             ;
7E54             ;"SEND" COUNTS LINE LENGTH AND FORCES
7E54             ;A CARRIAGE RETURN AFTER (MAXLEN) CHARACTERS
7E54
7E55 3A467D      PBYTE LD A,(ASTORE)
7E58
7E58             SEND PUSH HL
7E59 4F          LD L,A
7E5A 3A447D      S1 LD A,(MAXLEN); ;save char
7E5D 3C          INC A
7E5E 67          LD H,A
7E5F 3A4C7D      LD A,(PRHEAD) ;obtain max length+1
7E62 3C          INC A ;into H
7E63 324C7D      LD (PRHEAD),A ;last position
7E66 9C          CP H ;new position
7E67 2010        JR NZ,S2 ;check if over maxlength
7E69 3E0D        LD A,@DH ;if NZ then OK, goto S1
7E6B CD987E      CALL PRINT ;else send out carriage return
7E6E AF          XOR A
7E6F 324C7D      LD (PRHEAD),A ;and zero PRHEAD
7E72 7D          LD A,L
7E73 FE0D        CP @DH
7E75 200F        JR Z S4 ;never two carriage returns
7E77 18E1        JR S1
7E79 7D          S2 LD A,L
7E7A FE0D        CP @DH
7E7C 2005        JR NZ S3
7E7E AF          XOR A
7E7F 324C7D      LD (PRHEAD),A
7E82 7D          LD A,L
7E83 CD987E      S3 CALL PRINT
7E86 E1          S4 POP HL
7E87 C9          RET
7E87
7E88 A0A2A1A3     DB @0H,@2H,@4H,@6H,@8H,@AH,@CH,@EH,@FH,@H
7E8C A0A0A9AB     DB @0H,@2H,@4H,@6H,@8H,@AH,@CH,@EH,@FH,@H
7E90 A4A6A5A7     DB @0H,@2H,@4H,@6H,@8H,@AH,@CH,@EH,@FH,@H
7E94 AC4EADAF     DB @0H,@2H,@4H,@6H,@8H,@AH,@CH,@EH,@FH,@H
7E94
7E94             PRINT LD H,A
7E96 67          CALL PRINTER
7E99 CD1E7F      CP @DH ;printed a CR ?
7E9C FE0D        RET NZ
7E9E C9          LD A,(SW1)
7E9F 3A477D      LD A ;add linefeed(s)?
7EA2 B7          OR A
7EA3 C8          RET Z
7EA4 FE02        CP 2
7EA6 3E0A        LD A,@AH
7EA8 2003        JR NZ ONELF
7EAA CD1E7F      CALL PRINTER
7EAD CD1E7F      ONELF CALL PRINTER
7EB0 7C          LD A,H
7EB1 C9          RET
7EB1
7EB1             PBUF XOR A
7EB2 AF          LD (PRHEAD),A
7EB3 324C7D      LD A,@DH
7EB6 3E0D        LD (BUFFEND),A
7EB8 32FF5B      LD HL,BUFF-1
7EBB 21FF5A      JP SCAN
7EBB
7EBB             INIT LD HL,(START)
7EC1 2A487D      LD (POSIT),HL
7EC4 224D7D      LD A,(HL)
7EC7 AF          LD (PRHEAD),A
7ECB C9          RET
7ECB             TEST LD HL,(FINISH)
7ECC 2A4A7D      LD DE,(POSIT)
7EDF ED5B4D7D    OR A
7ED3 B7          SBC HL,DE
7ED4 ED52        RET
7ED6 C9          RET
7ED6
7ED6             PNMH CALL INIT
7ED7 CDC17E      LP1 CALL TEST
7EDA C0CC7E      JR Z EMPTY
7EDD 2013        EX DE,HL
7EDF EB          LD DE,LP1 ;return address
7EE0 11DA7E      PUSH DE ;onto stack
7EE3 05          LD A,(HEXD)
7EE4 3A457D      OR A
7EE7 B7          LD A,(HL)
7EEB 7E          INC HL
7EE9 23          LD (POSIT),HL
7EEA 224D7D      JR NZ MODE
7EED 2009        JP SEND
7EEF C3587E      LD A,@DH
7EF2 3E0D        JP SEND
7EF4 CD587E      CALL SEND
7EF7 C9          RET ;-----> TO BASIC
7EF7
7EF7             MODE PUSH AF
7EF8 F5          SRL A
7EF9 CB3F        SRL A
7EFB CB3F        SRL A
7EFD CB3F        SRL A
7EFF CB3F        CALL FORM
7F01 CD107F      POP AF
7F04 F1          CALL FORM
7F05 CD107F      CALL SPACE
7F08 CD4D7E      RET
7F0B C9          RET
7F0B
7F0C F1          BYT POP AF
7F0D C3587E      JP SEND
7F0D
7F10 E60F        FORM AND @FH
7F12 FE0A        CP @AH
7F14 3002        JR C F2
7F16 C007        ADD A,7
7F18 C607        ADD A,30H
7F1A CD587E      CALL SEND
7F1D C9          RET
7F1D
7F1E             ORG USR+1E0H
7F1E             LOAD USR+1E0H
7F1E
7F1E             PRINTER EQU 0
7F1E
7F1E
7F60 4100 7F1E 7F35 7F35 @0

```



```

1000 LET A = 32049
1010 CLEAR A
1020 LET A = A + 1
1030 READ D : POKE A,D
1040 IF A < 32587 THEN GOTO 1020
1042 PRINT "Start cassette"
1044 PAUSE 0
1050 SAVE "print" CODE 32050,549
1060 STOP
1070 REM MAIN PROGRAM DATA
1072 DATA
195, 79, 125, 195, 178, 126, 195, 215, 126, 195, 85, 126,
195, 85, 126, 195, 85, 126, 80, 0, 0, 0, 0, 0,
0, 0, 0, 0, 42, 83, 92, 34, 72, 125, 42,
75, 92, 34, 74, 125, 205, 193, 126, 205, 204, 126, 200,
26, 103, 19, 26, 111, 17, 48, 48, 1, 48, 48, 124,
181, 40, 31, 43, 12, 121, 254, 58, 32, 245, 62, 48,
79, 4, 120, 254, 58, 32, 236, 62, 48, 71, 28, 123,
254, 58, 32, 227, 62, 48, 95, 20, 24, 221, 122, 254,
48, 32, 2, 62, 32, 205, 88, 126, 123, 205, 88, 126,
120, 205, 88, 126, 121, 205, 88, 126, 42, 77, 125, 35,
35, 35, 17, 94, 125, 213, 35, 126, 254, 13, 32, 8,
205, 88, 126, 35, 34, 77, 125, 201, 254, 6, 32, 7,
62, 44, 205, 88, 126, 24, 231, 254, 14, 32, 7, 35,
35, 35, 35, 35, 24, 220, 254, 32, 48, 2, 24, 214,
24, 26, 0, 254, 128, 56, 21, 254, 144, 48, 17, 229,
22, 0, 95, 33, 8, 126, 183, 25, 126, 205, 88, 126,
225, 195, 176, 125, 254, 128, 56, 11, 254, 165, 48, 7,
62, 46, 205, 88, 126, 24, 171, 254, 255, 40, 8, 254,
165, 56, 46, 254, 255, 48, 42, 229, 205, 77, 126, 33,
150, 0, 214, 165, 40, 8, 71, 203, 126, 35, 40, 251,
16, 249, 126, 203, 127, 32, 6, 205, 88, 126, 35, 24,
245, 203, 191, 205, 88, 126, 205, 77, 126, 225, 195, 176,
125, 254, 127, 32, 8, 62, 99, 205, 88, 126, 195, 176,
125, 205, 88, 126, 195, 176, 125, 245, 62, 32, 205, 88,
126, 241, 201, 58, 70, 125, 229, 111, 58, 68, 125, 60,
103, 58, 76, 125, 60, 50, 76, 125, 188, 32, 16, 62,
13, 205, 152, 126, 175, 50, 76, 125, 125, 254, 13, 40,
15, 24, 225, 125, 254, 13, 32, 5, 175, 50, 76, 125,
125, 205, 152, 126, 225, 201, 160, 162, 161, 163, 168, 170,
169, 171, 164, 166, 165, 167, 172, 174, 173, 175, 103, 205,
30, 127, 254, 13, 192, 58, 71, 125, 183, 200, 254, 2,
62, 10, 32, 3, 205, 30, 127, 205, 30, 127, 124, 201,
175, 50, 76, 125, 62, 13, 50, 255, 91, 33, 255, 90,
195, 176, 125, 42, 72, 125, 34, 77, 125, 175, 50, 76,
125, 201, 42, 74, 125, 237, 91, 77, 125, 183, 237, 82,
201, 205, 193, 126, 205, 204, 126, 40, 19, 235, 17, 218,
126, 213, 58, 69, 125, 183, 126, 35, 34, 77, 125, 32,
9, 195, 88, 126, 62, 13, 205, 88, 126, 201, 245, 203,
63, 203, 63, 203, 63, 203, 63, 205, 16, 127, 241, 205,
16, 127, 205, 77, 126, 201, 241, 195, 88, 126, 230, 15,
254, 10, 56, 2, 198, 7, 198, 48, 205, 88, 126, 201
2090 REM PRINTER ROUTINE DATA
2091 REM FOR Z80PIO CHIP
2092 DATA
197, 245, 62, 207, 1, 31, 2, 237, 121, 62, 255, 237,
121, 62, 15, 1, 31, 3, 237, 121, 1, 31, 0, 237,
120, 203, 71, 32, 250, 241, 1, 31, 1, 203, 255, 237,
121, 203, 191, 237, 121, 203, 255, 237, 121, 193, 201

```

REM ZX ENERGY MANAGEMENT (C) 1983 A.D. CHAMIER

```

10 SLOW
20 FOR N=1 TO PI
30 LET L=64*Q+32*B+16*P+N
40 POKE K,L+8
50 FOR Q=1 TO 127
60 POKE K,L+4
70 POKE K,L
80 IF PEEK K>127 THEN NEXT Q
90 LET X(N)=LN ((64+Q)/(192-Q))

```

100 NEXT N

The familiar ZX81 print out—the quality's adequate but cannot match that produced by a Dot Matrix printer.

RANDOMIZE USR 32056 sends each byte to the printer, and this may cause some strange results if the file is anything other than a text file. However, with store 32069 set to non-zero, each byte is sent to the printer as a two byte hexadecimal number. With width set to 48, this gives a tabulation of memory, with the exception that addresses are not printed out.

PRINT A SINGLE CHARACTER

The code for the character is POKED into 32070 and LET A = USR 32059 will send it out. This is particularly useful for sending a control byte to the printer. If USR 32059 is going to be used repeatedly, then the print-head position counter at 32076 should be zeroed at the start of the sequence by a simple POKE from Basic.

Emulating WIDTH

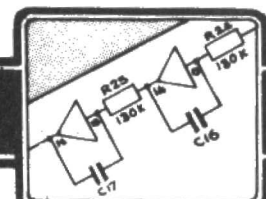
Store 32068 may hold a value between 1 and 255 which will represent the maximum number of characters to be printed on a line. It is set with a POKE from Basic.

CHANGES: The value in 32071 controls the line spacing of the printer. If your printer is hard-wired to operate a linefeed after every carriage return, then 0 should go in this store. A 1 or a 2 will, depending on your printer give you normal, double or triple spacing!

GRAPHICS: If you intend to drive an Epson 80 series printer with a full 8 bits of data using the Z80 PIO chip then you will be able to print the Spectrum's BLOCK graphics. In the main program POKE pulls into 32218 and 32219. In the printer routine POKE the following four bytes into locations 32575 to 32584 inclusive; 237, 121, 193, 201, 0. Characters which the printer cannot handle such as user-defined graphics are printed as a full-stop (46), but you may alter this to a symbol (or a space) of your choice by changing the value at location 32254.

Having now got your Spectrum printing nicely, there is every incentive to write a program to display in Hex the contents of memory, and to read the spare input lines on the PIO in a control application. At least the listing will be readable.

EE&CM



BEST STRAIGHT LINE

Paul Beverley and Steve Dowson with another article in their series of BBC micro applications. This month, the computer as a glorified calculator.

This month the topic under discussion is the use of the BBC microcomputer for analysing data which is suspected to be of a linear nature. That is to say, looking at two variables which are expected to have some sort of linear relationship to one another. If the data were analysed by hand, a graph of one variable against the other would be plotted, and then the best straight line through the points would be drawn by eye. Then by measuring the slope of the line and the intercept on the Y-axis, the equation for the straight line could be determined. There are numerical methods which can be used to calculate the best straight line using various statistical assumptions, but these require a fair amount of calculation. Programmable calculators have been used to a large extent for this application, and are very effective. This article shows, however, that it is possible to use the computer to get the best of both worlds. Not only does it do the numerical calculations for the slope and intercept, and the standard deviations of these quantities, but also the graphics facilities are used to plot the best straight line, so that its relation to the individual points can be seen.

There is another feature included which may not appeal to the purists amongst you. It is possible to go through the points one by one, and exclude each one of them from the calculation of the best straight line to see what difference it makes to the graph. The idea is that when looking at the graph one, or perhaps two points may be some way off the line. This could be because of a straightforward error in recording the values, or alternatively it could be a genuine reading which requires some sort of physical explanation for its abnormality. The program described enables a point that is some way off the line to be deleted and the difference time makes to the BESTFIT line to be noted. If it made a large difference it would be possible to go back to the experiment and check that particular reading to see if a mistake had been made.

Facilities offered

Let us look in general at the facilities which this program provides. First of all it allows the entry of pairs of X and Y values. Since this has to be done before anything else, when the program starts, it is in the mode where the user is prompted for X and Y values rather than going straight to the menu. Having entered these values, it is best to check through them to see if they are correct, and if necessary alter individual values. It is then possible to look at the graph which the computer draws, and if a printer with a screen dump facility is available, obtain a hard copy of the graph. Once the BESTFIT line is drawn it is possible to go through the points individually excluding them from the BESTFIT line as mentioned above. If one of the points is in fact incorrect and it is necessary to check it against the experiment it must be possible to find out which of the points was at fault. What the program can do therefore is to go back to the values that were entered and to sort them into

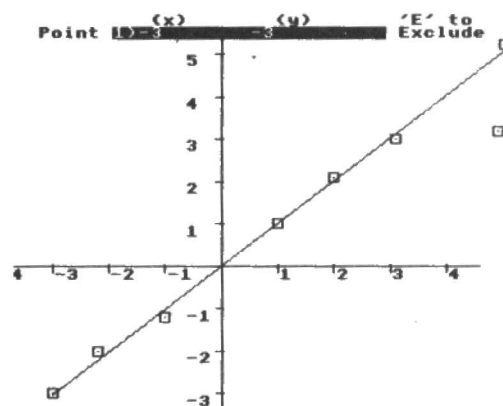


Figure 1. When the program starts it is in the mode in which the user is prompted for X and Y values.

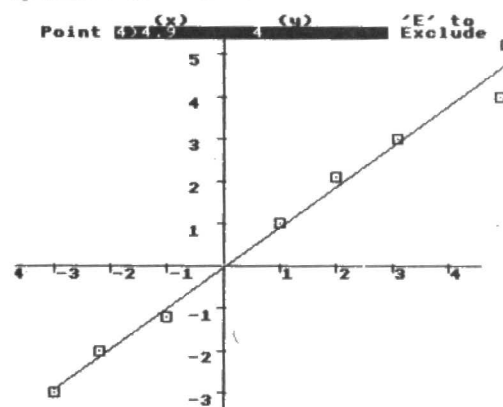


Figure 2. Once the BESTFIT line is drawn it is possible to see which, if any, of the points may be in error.

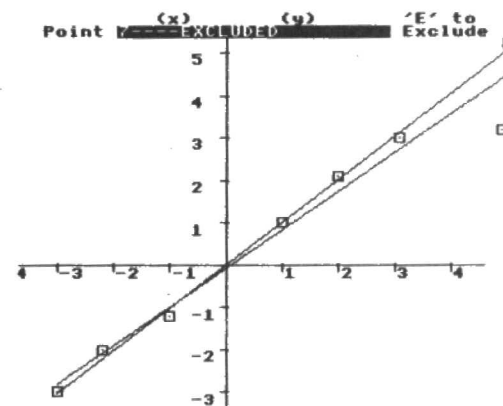


Figure 3. The program can sort the X values into numerical order. Having sorted the entries it is possible to exclude a point and note the effect on the BESTFIT line.



The other feature which has been put into this program is that of correlation. The idea of this is that if all the points lie exactly on a straight line and therefore the relationship is exactly linear then the correlation is unity. The more scattered the values are on either side of the line the smaller is the correlation.

```

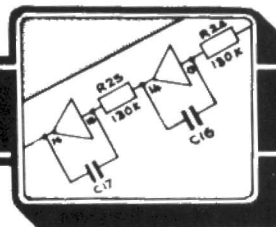
10 REM*****
20 REM* BESTLINE _ March 1983 *
30 REM* BBC version by Steve Dowson *
40 REM* Idea and good features all *
50 REM* due to Paul Beverley *
60 REM*****

70 REM Disc users omit REMS from here on
80 DIM X(51),Y(51),D(2)
90 N=0:fixed=0:EX=0
100 MODE1
110
VDU19,0,7,0,0,0,19,7,0,0,0,0,19,1,15,0,0,0,19,2,8,0,0,0
120 PROCdata
130 REPEAT
140 PROCmenu
150 IF B=1 CLS:PROCtable:PROCdata
160 IF B=2 PROCcompute
170 IF B=3 PROCdraw
180 IF B=4 PROCscale:PROCFix
190 IF B=5 THEN RUN
200 UNTIL B=6
210 END
220 REM*** END OF MAIN PROGRAM *****
230 DEFPROCcompute
240 IF N<3 PROCwarn:ENDPROC
250 PROCassume:IF yes THEN PROCcalc2 ELSE PROCcalcl
260 PROCresults
270 ENDPROC
280 REM*****
290 DEFPROCdraw
300 IF N<3 PROCwarn:ENDPROC
310 PROCscale
320 PROCassume:IF yes THEN PROCcalc2 ELSE PROCcalcl
330 PROCgraph
340 ENDPROC
350 REM*****
360 DEFPROCmenu
370 CLS
380 PRINTTAB(15,3)"M E N U"
390 PRINT""1.Check/change the points""2.Display values
for gradient,etc""3.Display the graph""4.Change size of
division marks""5.Start again""6.End"
400 B=VAL(GET$)
410 ENDPROC
420 REM*****
430 DEFPROCdata
440 PRINTTAB(0,27)" <RETURN> moves down the list"" D -
Delete point";SPC(4);"T - cursor to Top"" S - Sort on
X";SPC(7);"M - return to Menu"
450 MOVE0,170:DRAW1280,170:MOVE620,170:DRAW620,1024
460 pair=1:NUMBER=0:FLIP=0:new
=TRUE:PROCformat:PROCheading
470 ENTRY$="T"
480 REPEAT

```

JULY 1983

TECHNICAL FEATURE



```

1330 U=U/(N-2-A):V=SQR(U/D)
1340 W=SQR(U*(1/(N-A)+P/P/D)):CORR=G/SQR(E*H)
1350 ENDPROC
1360 REM*****
1370 DEFPROCcalc2
1380 IFEX=0 A=0 ELSE A=1
1390 IF EX=N B=1 ELSE B=0
1400 S=0:U=0:D=0:C=0:CORR=0
1410 FOR I%=1TON-B:IF I%=EX I%=I%+1
1420 S=S+X(I%)*Y(I%):D=D+X(I%)^2:NEXT
1430 S=S/D
1440 FOR I%=1TON-B:IF I%=EX I%=I%+1
1450 T=Y(I%)-S*X(I%):U=U+T*T:NEXT
1460 V=SQR(U/(N-1-A)/D)
1470 ENDPROC
1480 REM*****
1490 DEFPROCresults
1500 CLS:PRINTTAB(10,5)"Gradient = ";S
1510 PRINT"Standard deviation = ";V
1520 IF CORR<>0PRINTTAB(9,12)"Intercept = ";C:PRINT"Standard deviation = ";W:PRINTTAB(7,18)"Correlation = ";CORR
1530 A%=GET#
1540 ENDPROC
1550 REM*****
1560 DEFPROCscale
1570 xmax=X(1):ymax=Y(1):xmin=xmax:ymin=ymax
1580 FOR I%=2TON
1590 IFX(I%)>xmax THEN xmax=X(I%)
1600 IFX(I%)<xmin THEN xmin=X(I%)
1610 IFY(I%)>ymax THEN ymax=Y(I%)
1620 IFY(I%)<ymin THEN ymin=Y(I%)
1630 NEXT
1640 XMIN=xmin:YMIN=ymin:XMAX=xmax:YMAX=ymax
1650 IFxmin>0 THEN xmin=0
1660 IFxmax<0 THEN xmax=0
1670 IFymax<0 THEN ymax=0
1680 IFymin>0 THEN ymin=0
1690 xsc=1164/(xmax-xmin):ysc=904/(ymax-ymin)
1700 xoff=100-xmin*xsc:IF xoff<100THEN xoff=100
1710 yoff=40-ymin*ysc:IF yoff<40THEN yoff=40
1720 side=90:IF xmax=0 THEN xoff=1164:side=-10
1730 IF ymax=0 THEN yoff=904
1740 IF NOT fixed PROCdiv(xmax-xmin,1):PROCdiv(ymax-ymin,2)
1750 ENDPROC
1760 REM*****
1770 DEFPROCgraph
1780 CLS:VDU5
1790 MOVExoff,0:DRAWxoff,960:MOVE0,yoff:DRAW1200,yoff
1800 PROCmarkx(D(1)):PROCmarkx(-D(1))
1810 PROCmarky(D(2)):PROCmarky(-D(2))
1820 FOR I%=1TON
1830 PROCpoint(I%)
1840 NEXT
1850 PROCline(S,C)
1860 VDU4
1870 PRINTTAB(11,0);"(X)";SPC(7);"(Y) 'E' to"
1880 PRINTTAB(2,1)"Point";SPC(24);"Exclude"
1890 VDU4,28,8,1,29,1,17,131,17,0
1900 ST=S:STC=C:pair=1:CLS
1910 REPEAT

```

```

GCOL0,1:PROCpoint(pair):PRINT"pair;";X(pair):TAB(11);Y
1r);
1920 A=GET
1930 IF A=13 GCOL0,3:PROCpoint(pair):pair=pair+1
1940 IF A=69 AND N>3 PROCtest
1950 IF A=69 AND N=3PRINT"Not enough points":A%=GET
1960 IF A=84 GCOL0,3:PROCpoint(pair):pair=1
1970 IF pair=N+1 THEN GCOL0,3:PROCpoint(N):pair=1
1980 UNTILA=77
1990 GCOL0,3
2000 VDU26,17,128,17,3
2010 ENDPROC
2020 REM*****
2030 DEFPROCpoint(pair)
2040 PLOT69,X(pair)*xsc+xoff,Y(pair)*ysc+yoff

```

```

2050
VDU25,0,-12;12;25,1,24;0;25,1,0;-24;25,1,-24;0;25,1,0;24;
2060 ENDPROC
2070 REM*****
2080 DEFPROCdiv(size,axis):GOTO2110
2090 H=size/10
2100 J=ABS(H):IFJ<1 THEN J=J/10
2110 d=10^(INT(LOG(size))):try=size/d;q=1
2120 IFtry<5THEN q=2:IF try<2 THEN q=5:IFtry<1 THEN q=10
2130 D(axis)=d/q
2140 ENDPROC
2150 REM*****
2160 DEFPROCmarkx(D)
2170 LOCALN
2180 N=0:REPEAT

```

```

N=N+D:R=N*xsc+xoff:MOVER+5,yoff-10:PRINT;N:MOVER,yoff-16:PLD
T1,0,20:UNTILR>1100 OR R<0
2190 ENDPROC
2200 REM*****
2210 DEFPROCmarky(D)
2220 LOCALN
2230 N=0:REPEAT
N=N+D:R=N*ysc+yoff:MOVExoff-(side),R-5:PRINT;N:MOVExoff-16,R
:;PLOT11,20,0:UNTILR>900 OR R<0
2240 ENDPROC
2250 REM*****
2260 DEFPROCline(S,C)
2270
xbase=xmin*xsc:ybase=(S*xmin+C)*ysc:MOVExbase+xoff,ybase+yof
f
2280
xtop=xmax*xsc:ytot=(S*xmax+C)*ysc:DRAWxtop+xoff,ytot+yoff
2290 ENDPROC
2300 REM*****
2310 DEFPROCwarn
2320 PRINTTAB(1,26)"Least squares method will not
work"with less than 3 points"

```

```

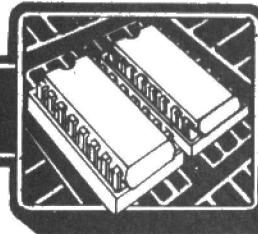
2330 A%=GET#
2340 ENDPROC
2350 REM*****
2360 DEFPROCsort
2370 FOR I%=1 TO N-1
2380 FOR J%=I% TO N
2390 IF X(I%)>X(J%)THEN
ST=X(I%):X(I%)=X(J%):X(J%)=ST:ST=Y(I%):Y(I%)=Y(J%):Y(J%)=ST
2400 NEXTJ,I%
2410 ENDPROC
2420 REM*****
2430 DEFPROCtest
2440 EX=pair:PRINT"EX: "----EXCLUDED ";
2450 IFyes PROCcalc2 ELSE PROCcalc1
2460 GCOL0,2:PROCline(S,C)
2470
A%=GET#GCOL0,0:PROCline(S,C):GCOL0,3:PROCline(ST,STC):EX=0
2480 CLS
2490 ENDPROC
2500 REM*****
2510 DEFPROCfix
2520 CLS
2530 PRINTTAB(4,4)"SCALE DIVISIONS - CURRENT VALUES"
2540 PRINTTAB(18,10)"X-AXIS Y-AXIS"
2550 PRINT"Max value: ",XMAX,YMAX"Min value: ",XMIN,YMIN
2560 PRINT"Current"divisions: ",D(1),D(2)"Enter
new"values...";
2570 REPEAT:T=18
2580 FOR I%=1TO2:INPUTTAB(T,20),AX$:PROCvalid(AX%):IF
NUMBER THEN D(I%)=VAL(AX$)
2590 T=28:PRINTTAB(5,22)" "NEXT
2600 PRINTTAB(5,22)"OK?":A%=LEFT$(GET#,1)
2610 UNTIL A%="Y"
2620 IFNUMBER fixed=TRUE
2630 ENDPROC

```

Finally there are one or two comments to make about the listing of the program itself. The program was originally written for the Atom computer, but was translated for the BBC microcomputer by Steve Dowson. It demonstrates very good use of long variable names and of procedures in order to give the program a good structure and to make it easier for other people to read. Also, anybody wanting to tailor it for their applications has a program that is easy to amend. When using a disc system, in order to make sufficient space for the program plus variables and still to be able to work in mode 1, it will be necessary before loading the program, to change the value of PAGE to &1200. Although the full listing of the program is given with this article, a cassette of the program to save the effort of typing the program in and trying to find all your typing mistakes is available at a cost of £2.00 from:

Steve Dowson,
"Roots",
Poorhouse Lane,
Bracken Ash,
Norwich,
NR14 8EN

E&CM



BBC Sequencer Interface

Robert Penfold describes an interface that allows the BBC micro to control a synthesiser.



Many monophonic synthesisers have some form of built-in digital control system these days, and this often includes some kind of sequencer. A sequencer greatly enhances the usefulness of the instrument by enabling an inexperienced player to program the instrument so that the desired melody is played in note perfect fashion with precision timing, or by using a short loop sequence the unit can function as a simple rhythm machine.

Unfortunately, not all synthesisers have a built-in sequencer, and those that are equipped with one usually have a fairly simple type. For example, the author's SCI Pro One synthesiser has an integral one hundred note sequencer, but the notes must be of the same duration, and there is no way of controlling the gate pulse duration.

The idea of using a home-micro to control a synthesiser is by no means a new one, and it is in fact a well tried technique which normally gives excellent results in practice. This article describes a simple interface which enables the BBC Model B micro to be used as a versatile sequencer for the ever popular SCI Pro One, or any other synthesiser which has the standard one volt per octave control voltage input plus a 5 volt positive gate/trigger input.

Block Diagram

In principle the sequencer interface is straightforward, as can be seen from the block diagram of **Fig 1**. The control voltage is obtained using a digital to analogue converter which is fed from lines PB0 to PB5 of the BBC micro's user port. The digital to analogue converter is in fact an eight bit type with the two least significant bits connected to the 0V rail. This gives 63 programmable output voltages plus zero volts, and the use of a good quality eight bit converter gives excellent accuracy and stability.

The keyboard voltages of a synthesiser are normally provided by a multi-stage potential divider circuit which consists of a long series of precision resistors of the same nominal value. Thus, like the

digital to analogue converter, the keyboard provides a series of voltages that rise in equal steps. In order to give the correct notes using the sequencer it is merely necessary to match the size of the voltage steps to those provided by the keyboard, and this is accomplished using a variable gain amplifier at the output of the converter. This also acts as a buffer stage which gives the unit a suitably low output impedance.

With twelve notes per octave (including semitones) and 63 possible control voltages a range of just over five octaves can be covered. Most synthesisers have about a three octave keyboard, but the circuitry is normally capable of a much wider frequency range than this, and most instruments are capable of tracking accurately over the full five-and-a-bit octave control range.

Apart from providing a control voltage the unit must provide a trigger pulse if the synthesiser has an AD (attack-decay) envelope shaper, or a gate pulse if it has the more common ADSR (attack-decay-sustain-release) type. There are two ways of providing the necessary signal with the system used here. The first of these is to program the output port so that the CB2 line of the user port becomes an output which briefly pulses low each time the computer sends data to the port. This pulse is used to trigger a monostable multivibrator which produces a trigger/gate pulse of preset duration.

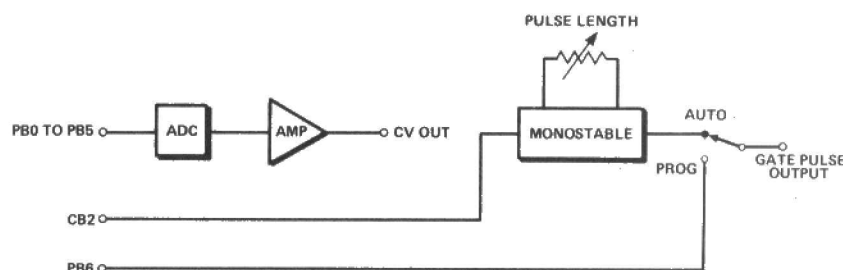
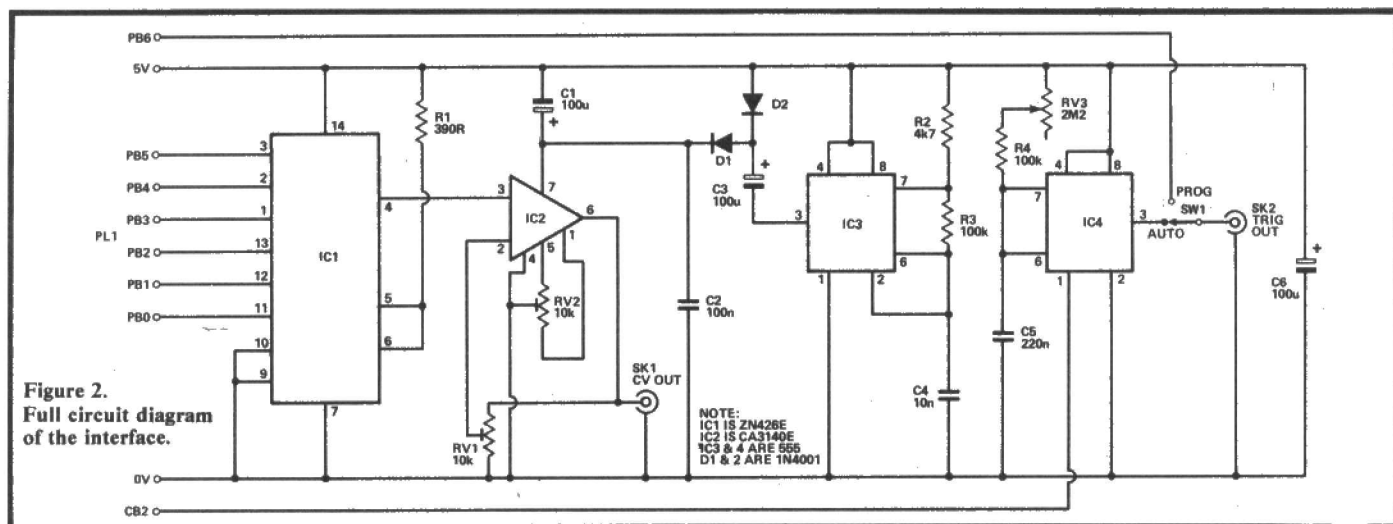
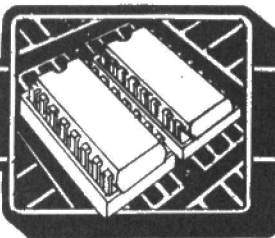


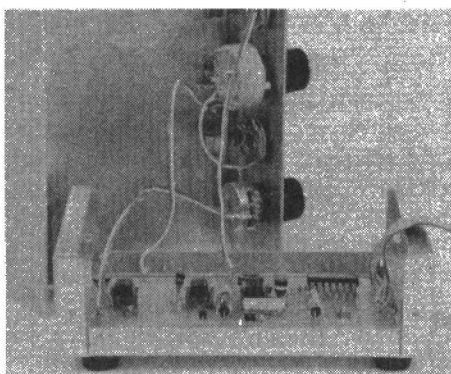
Figure 1. The block diagram of the sequencer interface shows that the design is based on a few basic building blocks.



This system automatically provides the trigger/gate pulse each time a new note is sent to the output port, or even if a note is repeated since it is writing to the port that generates the pulse, and not a change on the data lines of the port. For those who are not familiar with the 6522 PIA which is used to produce the user (and parallel printer) ports of the BBC model B computer, it should perhaps be explained that the CB2 line of the output port is controlled by the peripheral control register of the 6522. In the BBC micro this is at address &FE6C incidentally. It is bits 5 to 7 of this eight bit register that control the operating mode of line CB2, and for the pulse mode of operation these bits should be at 1, 0, and 1 respectively, or, to put it another way, this register should be set at 160 (128 plus 32). Therefore, in order to obtain this pulsed output mode it is merely necessary to include `?&FE6C=160` (the BBC micro equivalent of POKE hex address FE6C to 160) as a program line near the beginning of the program.

The second method of gating/triggering does not require any additional hardware, but is a little more complex in that the software has to accommodate a third parameter in addition to pitch and note duration. This method is simply to use line PB6 to provide the gate/trigger signal, and using suitable software the output pulse duration can be different for each note if desired. This obviously gives great flexibility when used with a synthesiser which has an ADSR envelope shaper, but it takes longer to

program the system with the required melody or rhythm. For this reason it is better to use the automatic triggering mode unless the flexibility of programmed gating is really needed.



An internal view of the completed Sequencer Interface.

Construction

A suitable printed circuit design for the interface is shown in Fig 5. IC2 has a PMOS input stage and in order to minimise the risk of damage by high static charges it should be the last component to be fitted to the board and it should be fitted in an 8 pin DIL socket. Leave this device in its protective packaging until it is to be fitted on the board, and handle it as little as possible. Do not overlook the four link wires and fit Veropins at points on the board where connections to off board components such as S1 and VR3 will be made.

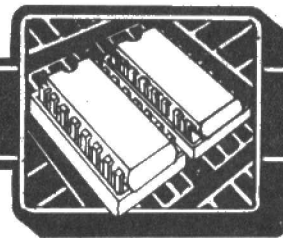
The connection to the computer is made via a 20 way ribbon cable fitted with a 20 way IDC header socket. It is advisable to buy this as a socket and lead

ready-wired together. When connecting this cable to the printed circuit board be careful to get the cable the right way round, and make sure that none of the leads are accidentally swapped over.

An aluminium box measuring about 133 x 102 x 38mm makes an inexpensive but reasonably smart housing for the interface. The controls and sockets are mounted on one of the 133 x 38mm sides of the case which then acts as the front panel, and the printed circuit board is mounted on the base panel towards the rear of the case. Spacers are used over the mounting bolts for the board to ensure that the connections on the underside cannot come into contact with the metal casing. The ribbon cable is thin enough to go between the base and top sections of the case without causing any problems, and it is not necessary to cut an exit slit for this lead.

Adjustment

With PL1 connected to the user port of the BBC micro, the SK1 plus SK2 connected to the appropriate sockets of the synthesiser using the usual jack leads, the next step is to set up lines PB0 to PB6 on the user port as outputs. This display entails typing `?&FE62=127` into the micro and hitting the "return" key. Next the output port is set at a number which is equal to the number of keys on the synthesiser, which is 37 for a normal three octave type. This is achieved by typing `?&FE60=37` and then hitting "return". RV1 should now be adjusted so that the control voltage



The circuit

Figure 2 shows the full circuit diagram of the Sequencer Interface.

IC1 is a Ferranti ZN4267E digital to analogue converter which gives good performance, will operate from the single 5 volt supply available from the BBC micro's user port, and is reasonably inexpensive. Pins 9 and 10 of IC1 are the two least significant bit inputs, and as explained earlier, these are simply tied to the 0V rail. The internal 2.55 volt (nominal) voltage reference of IC1 is utilised simply by wiring pins 5 and 6 together and adding discrete load resistor R1. A 1u decoupling capacitor can be added from pins 5 and 6 to the 0V rail, but this does not seem to be necessary in the present application.

The amplifier is a straightforward non-inverting type based on operational amplifier IC2. A CA3140E is used because it has a class A output stage which enables its output voltage to swing down to virtually the 0V supply potential. This avoids having to provide a negative supply rail.

RV1 controls the voltage gain of the amplifier, and in practice this will be set to give a gain of only about two times. RV2 is the offset null control which is used to compensate for small offset voltages in both IC1 and IC2 so that accurate results are obtained at low output voltages.

One slight problem with this arrangement is that IC2 is unable to produce an adequate maximum output voltage. With a range of just over five octaves and one volt per octave required, IC2 must give a maximum output potential of just over 5 volts. With a nominal 5 volt supply this is clearly impossible, but in actual fact the output of IC2 cannot go within a volt or two of the positive supply potential and the maximum output voltage of this device would fall well short of the desired level.

A simple and inexpensive solution to the problem is the 555 timer IC3 as an oscillator which feeds a rectifier and smoothing circuit. The latter gives a positive output voltage which is about 2.5 volts above the positive supply rail. This gives a potential of about 7.5 volts above the 0V rail, and

this is adequate to give the required voltage swing at the output of IC2.

The monostable used in the automatic triggering mode is based on another 555 timer IC. This requires a very brief trigger pulse at pin 2 which must take the trigger input below one third of the supply potential. As the output pulse on the CB2 line is a very short negative pulse which goes virtually down to the 0V rail potential it can be used to directly drive the input of IC4. VR3 enables the output pulse length to be varied from about 24 milliseconds at minimum resistance to approximately 557 milliseconds at maximum resistance. This range should be adequate, but the time component values could be modified if a different timing range is required.

S1 connects the trigger output socket (SK2) to either the output of IC4 or the PB6 line, as required. In both cases the signal obtained is a pulse waveform virtually equal to the 5 volt supply potential.

from the interface gives the same note that is obtained using the top key of the keyboard, and it is very easy to match the two notes if you frequently switch between the keyboard and the external control voltage. The "drone" switch of the synthesiser can be used to hold the envelope shaper open while making this adjustment.

Next the output port should be set at one by typing ?&FE60=1 and hitting return. RV2 is then adjusted to match the note obtained using the interface to that obtained using the lowest key of the keyboard. It is then advisable to repeat adjustments of RV1 and RV2 a few times until the interface is exactly in tune with the keyboard and no further improvement can be made. You should then find that setting the output port at numbers between 1 and 37 gives the same note as the corresponding key of the keyboard. Numbers up to 63 will

provide additional notes which are not available using the keyboard. Setting the output port at zero will probably not cut off the VCO or VCOs in the synthesiser, but if it does this represents an easy way to put gaps between notes. The other way of achieving this is to use the programmable gating mode, or simply to make notes longer than the duration which the envelope shaper will provide.

The equipment is ready for use once VR1 and VR2 have been set up correctly, and a few simple programs which can be used with the unit are given below.

It is possible to load notes from the synthesiser's keyboard into the BBC micro using the simple arrangement shown in **Fig 3**. Here the control voltage from the synthesiser is fed via variable attenuator VR4 to the channel 1 input of the BBC micro's analogue port. VR4 keeps the voltage fed to the channel 1 input within the 1.8 volt maximum, and

enables the system to be adjusted so that the correct number is produced by each key. $X\% = \text{ADVAL}(1)/1000+1$ returns the required value (it was found to be necessary to include the "+1" to compensate for a small amount of non-linearity somewhere in the system). The simple program shown below displays the returned value so that RV4 can be set correctly by empirical means.

```
10 X% = ADVAL(1)/1000+1
20 PRINT TAB(10,10)X%
30 GOTO 10
```

This program is useful for converting keyboard notes to numbers for use in sequencer programs, but it would presumably be possible to devise software that would enable notes to be directly loaded into a program from the synthesiser's keyboard.

The simple program given below is intended for use where a short loop is required to enable the system to be used as a rhythm machine. It gives maximum versatility by making use of the programmed gate pulse facility of the interface. The pitch, gate time, and total note duration are placed (in that order) in the data statement at line 140, and a single data line is all that should be needed since up to 255 characters can be used here. However, note that the number of notes in the data statement must agree with the figure at the end of line 40. The gate pulse and note duration times are in centiseconds incidentally. The program can be halted by operating the spacebar.

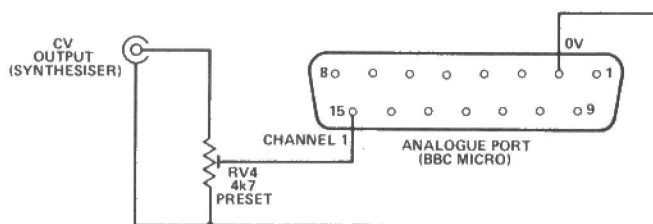
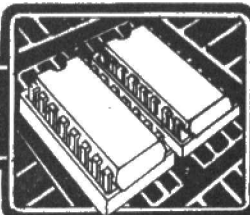


Figure 3. Circuit diagram of the connections that enable notes from the synthesiser's keyboard to be loaded into the BBC.



```

10 ?&FE62 = 127
20 REPEAT
30 RESTORE
40 FOR play = 1 TO (number of
   notes at line 140)
50 READ pitch, gate, duration
60 ?&FE60 = pitch+64
70 T=TIME
80 REPEAT UNTIL TIME=
   T+gate
90 ?&FE60=pitch
100 REPEAT UNTIL TIME=
   T+duration
110 Next play
120 UNTIL INKEY(0)=32
130 ?&FE60=0
140 Data pitch, gate, duration, pitch,
   gate, duration, etc.

```

The program given (left) is intended for use with the auto-triggering mode and enables long sequences to be easily programmed, edited, and stored.

Program Notes

The program is menu-driven and has facilities for loading music into the storage array (set up in line 50) either direct from the keyboard or from a previously-saved tape. It is possible to add to music already in the array, either from the keyboard or from tape. The facility for saving music onto tape can record the whole or a selected part of the music in the array.

There are two 'play' facilities, the first playing the music through once only, the

second playing it as a loop, continuously, for as long as required. While the music is being played, the 'menu' remains on the screen, but the option chosen changes colour, to indicate what is happening.

Option 3 is a powerful editing facility. Really it is three options in one. Firstly, any note can be called by number and displayed on the screen, and the pitch and/or time altered if required. Secondly the tempo of the music can be made faster or slower by multiplying or dividing all the time values. Thirdly, there is an option to delete any note, or to insert a note at any position.

The program has been made very 'robust', so as to be usable by people not familiar with computers. All input lines are fitted with 'garbage filters', and the program responds to 'silly' input by beeping and returning to the input line or the 'menu', as appropriate. Pressing 'Escape' returns the program to the menu.

Figure 4. The sequencer interface's overlay is shown below.

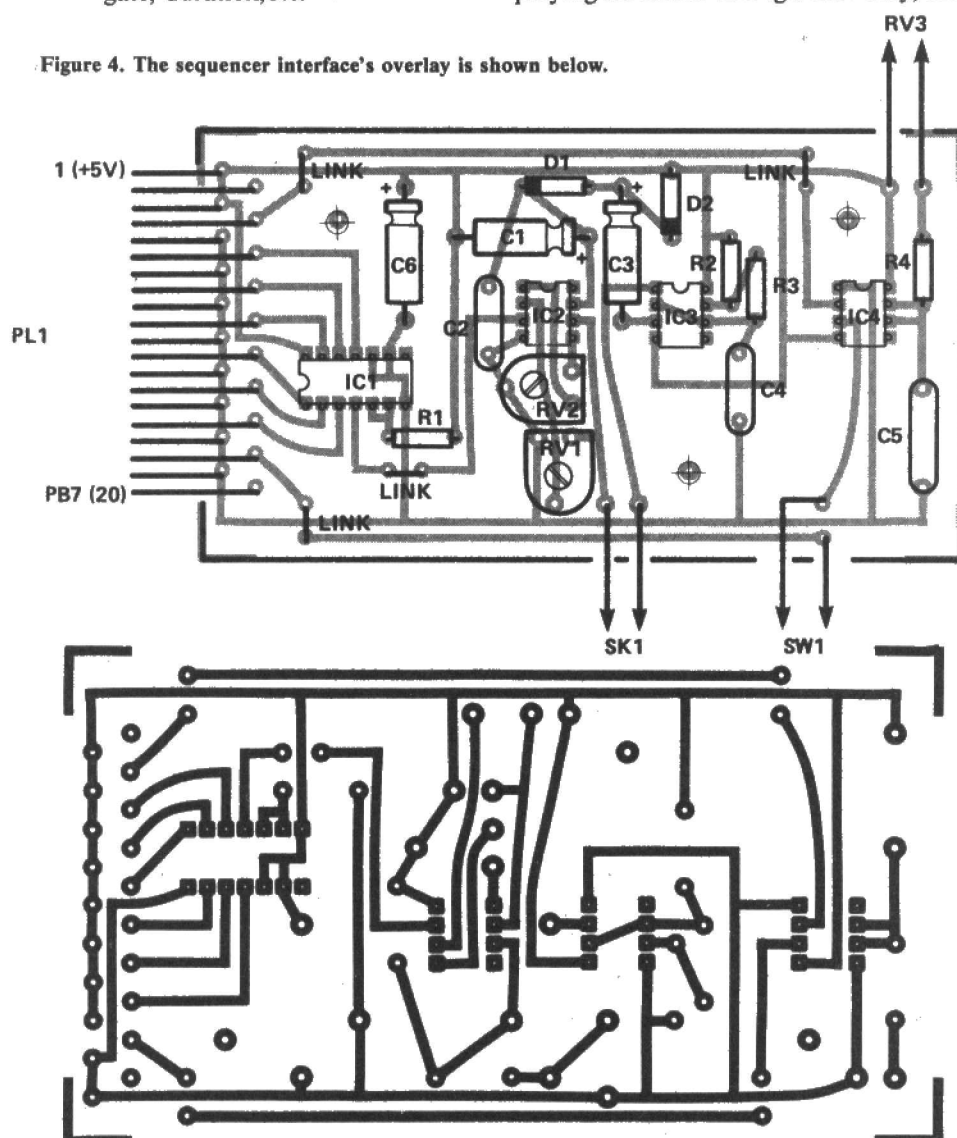


Figure 5. The PCB foil pattern shown full size.

Next Month - Full software listings.

Parts List

Resistors, all 1/4 watt 5%

R1	390R
R2	4k7
R3, 4	100k (2 off)

Potentiometers

RV1, 2	10k 0.1 watt horizontal preset
RV3	2M2 linear carbon

Capacitors

C1, 3, 6	100u 10V axial electrolytic
C2	100n polyester
C4	10n polyester
C5	220n polyester

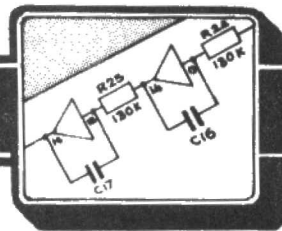
Semiconductors

IC1	ZN426E
IC2	CA3140E
IC3, 4	555
D1, 2	1N4001

Miscellaneous

SK1, 2 6.35mm jack sockets (2 off), S1 SPDT toggle switch, PL1 20 way IDC socket with cable, Case, PCB, control knob, etc.

E&CM



THE COMPUTER BRAIN

Mike James continues his series dealing with artificial intelligence with an article he's called 'Fuzzy Programs'.

The idea of a computer program that reasons by using rules of the IF condition THEN conclusion type, that we met in last month's Computer Brain on expert systems, seems very natural and promising until you start to consider the sort of judgments humans make. It is not very often that a decision can be made with absolute certainty. It is more often the case that we say things like – "I think the trouble might be" or "It could be . . ." than we say "I Know" or "It is". Last month's expert system assumed that the answer to any question was known with absolute certainty and that the presence of any given condition was always an absolute sign that the conclusion should be drawn. It is certainly clear that this is not always so! A component of human reasoning that we have ignored so far is uncertainty.

To be, or not

There are two ways of dealing with uncertainty in reasoning. The first is traditional and based on probability, the second is fairly new and not so well developed so there is scope for experimentation. It is important however, to realise that there are two different sorts of uncertainty that occur in reasoning. The first is just not being sure of the condition. For example, you might have caught sight of an animal only for a moment and not be sure if it had a tail or not. This is being uncertain of the evidence and is generally easy to deal with. The second type of uncertainty is where the evidence is quite clear cut, that is you are sure of the facts, but there is no certain connection between what you know and the conclusion that you draw. For example, you may be sure that the weather is humid but this only increases the chances that there will be a thunder storm. This form of uncertainty is known as 'uncertainty in inference' and is the subject that will occupy most of this article.

Probability

We are all familiar with the idea of the probability of something happening. Even so, most of us have only an intuitive idea that the higher the probability the more likely the thing is to happen. A probability of zero corresponds to the certain knowledge that an event will not happen and a probability of 1 corresponds to the certain knowledge that an event will happen. So probabilities of 0 and 1 correspond to certainty and values in between correspond to uncertainty. (You might notice the similarity here between probability and boolean logic). The best interpretation that we have of probability is in terms of the number of times something happens. For example, if you toss a coin a great many times it will come up heads roughly half of them. In this sense we can say that the statement 'the probability of getting heads is 0.5' is a statement about what proportion of heads we would expect to see in a large number of throws. This idea generalises to an interpretation of the probability of other events as the proportion of the time that they occur in the long run.

This all sounds like a good solid realistic way to understand

probability and it is generally considered to be the best. Consider, however, the situation where you are asked to say how likely something is to be true or false. In this case it is difficult to see how the idea of the number of times something is true or false in the long run can be used as an interpretation of the probability. For example, 'what do you reckon the probability is that life exists on other planets?' is a question that you might be able to answer. It is difficult though, to see how the probability that you give as an answer can be interpreted as the number of times you are likely to be right in the long run. Even the wildest imagination will find it difficult to cope with more than one universe in which to repeat the event! There are many ways round this difficulty, but the easiest is to abandon the direct interpretation of probability as an indication of how often something would occur. Of course once you do this you can only use probability as a vague measure of how certain you are of something and this is not something that can be verified in the same way as the proportion of times that something happens – it is in this sense subjective. The trouble is that once you abandon the physical meaning of probability there is nothing to recommend it over and above any other measure of uncertainty!

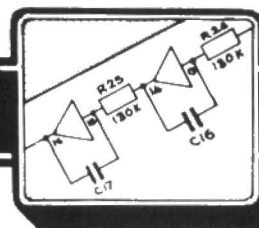
Probably uncertain

Rather than use probability you could choose to work with another measure of uncertainty with a different set of rules for combining such measure and because the whole thing is subjective no one could argue with you! The point is that while probability may be just right for summarising our uncertainty about some events in the world it isn't necessarily the best way of summarising the way humans 'feel' uncertain about something. You may be feeling a little sceptical about using something other than probability, so in the second part of this article we take a look at one such alternative, fuzzy logic.

The laws of uncertain thought?

Even though there are philosophical problems when using probability in some areas of human knowledge and reasoning, it is still the system that we are most used to. For this reason it is worth examining how probability could be added to the sort of expert system discussed last month. To be able to do this we need to look at a little of the theory of probability. The probability of an event x is normally written as $P(x)$ which should be read as a shorthand for 'the probability of x '. For example, $P(\text{heads})$, is the probability of getting heads when a coin is spun, i.e. 0.5 if the coin is 'fair'. A slightly more complicated but very useful idea is that of 'conditional probability'. This is usually written as $P(A|B)$ and is read as 'the probability that A will happen given B has happened'. (You will find this article easier to read if you get into the habit of reading symbols like $P(A|B)$ as 'the probability of A given B '. That is, read P as 'the probability' and the vertical bar as 'given'). So for example, $P(\text{rain})$ is just the probability of rain





prisingly simple and surprisingly unsatisfactory. If you include in an expert system a supplementary question such as, "How sure are you (on a scale 0 to 1) of your last answer?" then you will collect estimates of the certainty of each piece of information that you are going to base your conclusion on. If you treat these estimates as probabilities then the correct way of using them to estimate the uncertainty in your conclusion is rather complicated and depends on knowing the relationships that exist between the different pieces of information. As these relationships are generally unknown, the usual assumption is that they are non-existent and this implies that the correct estimate of the uncertainty of the conclusion should be obtained by multiplying all the probabilities together. (This is because of a theorem in probability that says that if a number of events are independent the probability of them all happening is the product of their individual probabilities). This sounds very reasonable until you notice that .5 times .5 is rather small and even if you start out with large certainties, such as .9 the final certainty of the conclusion will be very small if you have a lot of evidence! Multiplying probabilities gives a very conservative estimate of the certainty of the conclusion because we are ignoring the relationship between the evidence!

There is no really acceptable solution to this problem and most expert systems invent their own way of dealing with it. However there is one method that is appealing because it fits in with the topic of our next section, fuzzy logic. Instead of multiplying the certainties together the certainty of the conclusion is estimated by taking the minimum of the certainties.

Alternatives to probability

You may have noticed that although we started out by being exact and using probabilities in precise ways, the section on uncertain thought introduced the idea that it's not so much the exact values that matter more the way that they show roughly how a piece of evidence supports or denies a conclusion. In the last section we saw that it was better to not use probability methods in estimating the certainty of a conclusion. In fact once you start working with beliefs and evidence there is nothing to say that you have to use probability at all! An alternative system is provided by fuzzy logic which is easiest to explain from the point of view of adjectives. If you think about the word 'tall' and how you use it, you will realise that there is an element of uncertainty in how you apply it. It's not that someone has a probability of being tall rather than they are more or less tall. A 7ft. person is almost certainly tall but what about someone 6ft 6ins, or just 6ft or 5ft 9ins? Tall is clearly not something that can be easily distinguished by a yes/no type decision, it's more vague than that. In traditional logic we use 1 to mean true and 0 to mean false and there are no values in between. In fuzzy logic you can use 'truth values' between 0 and 1 and once again 0 means false and 1 means true and intermediate values indicate 'more or less true or false'. For example, the answer to the question 'is he tall?' might not be a clear cut 0 or 1 (i.e. yes or no) but .5 or .8 or some such intermediate value.

This idea of using intermediate truth values seems a little strange at first but it doesn't take too long to get used to it. The fundamental logic operations AND, OR and NOT generalise

quite quickly to fuzzy truth values –

A AND B becomes MAX(A,B)

A OR B becomes MIN(A,B)

and NOT (A) becomes $1-A$

irrespective of whether A and B are good old boolean 0/1 truth values or fuzzy logic. In other words, even in traditional logic, $A \text{ AND } B = \text{MIN}(A, B)$:

A	B	A AND B	MIN(A,B)
0	0	0	0
0	1	0	0
1	0	0	0
1	1	1	1

This means that we can work with fuzzy logic and if the truth values happen to be zero and one we are doing nothing new – obviously traditional logic must be a subset of any new sort of reasoning that we are planning to use. From these simple beginnings we can go on to develop the whole of logic but now using fuzzy truth values.

There are many aspects of traditional logic that do not translate straight into fuzzy logic. This means that you cannot just replace traditional logic by fuzzy logic in any given application and to date no-one has really managed to work it into an expert system in any really convincing way – but it's early days yet!

One of the exciting uses for fuzzy logic is the description of human ideas to a computer. For example, the adjective 'tall' that was giving us so much trouble earlier could be described by a graph showing the truth value for various heights. This description is just as easy to convey to a computer as to another human. As well as simple descriptions of words such as 'tall', fuzzy logic can be used to convey vague rules such as 'if the boiler is hot' THEN 'turn the heat down' or IF 'the boiler is very hot' THEN 'turn the heat down a lot'. The words 'hot', 'very hot' and 'a lot' are all difficult to describe to a computer unless you are aware of the idea of fuzzy logic. This is one idea that there have been a few applications of, such as controlling a steam engine and cement kiln! You may have noticed that we have now come full circle. The early parts of the computer brain were concerned with the idea of a heuristic rule as opposed to an algorithm. Heuristics are vague rules that TEND to give you a solution rather than *always* give you a solution. What better way of describing a heuristic than a fuzzy rule of the type given above!

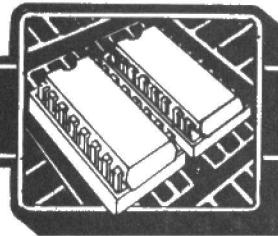
Conclusion

This month's computer brain has not included any programs because the application of the ideas of probability or fuzzy logic would require rather more space than available. This is not to say that writing programs using either idea is difficult but to find a good convincing example you have to work with a real application. However, this article should have provided you with enough information to write your own program to analyse faults in cars, or even computers!, using measures of uncertainty of both evidence and inference. This is both an exciting and potentially profitable area of A.I. so be persistent!

Next month is the last part of the Computer Brain and it is time to look back, draw some conclusions and have fun with one last 'intelligent' program. E&CM

E&CM

JULY 1983



Floppy Disc Controller

Paul Izod and Alan Stirling describe the floppy controller card for our Hi-Res computer project. This is the board that brings the FLEX operating system to life.



The floppy disc controller card is the last of the major boards that go to make up our Hi-Res computer. With the boards so far described and a suitable disc drive unit, the result is a powerful 6809 based system capable of supporting the powerful FLEX OS.

Board overview

This board interfaces up to 4 disc drives to the system. They may be any combination of 5¼" or 8", single or double sided. 5¼" drives may be single or double density, although with a CPU clock rate of 1MHz, 8" drives are limited to single density. The board however has been designed with 2MHz operation in mind, so it should be possible to use it with 8" double density drives.

There are five spare IC positions on the board to allow for customisation or extra facilities.

The board also has its own oscillator, although the backplane clock may be used instead.

The WD2793 in brief

The floppy disc controller board is built around the Western Digital 2793 Controller/Formatter. This device is the interface between the system bus and the

disc drive control and data signals. It is similar to its predecessors, the FD179X family but has on-chip phase lock loop data separator, and write precompensation logic.

A test mode is selected by resetting the 2793 and then grounding pin 22. This brings out internal timing signals for setting up purposes. R3, R4 and C5 are then adjusted.

The 2793 won't be described further here. Those interested should obtain a data sheet.

For a more general description of controller chips and their operation, "An Introduction to Microcomputers Volume 3" by Gerry Kane, is a useful reference.

Assembling the board

The PCB has a legend to show the position of all components and links, although as it doesn't give the value of the components, reference to the components list will be necessary.

It is recommended that a socket be used for 2793 – take care with its very fragile legs. Whether or not sockets are used for the other ICs is a matter of personal preference. If you decide to use them though, turned-pin sockets are worth the extra money. It is suggested that dual-in-line switches are used in link positions A, B, C and F.

If the backplane clock is to be used for the 2793 then carefully cut the track marked INT-CLK (above IC10) and add a short link EXT-CLK.

Construction is now straightforward. If sockets are used, then solder these in first, followed by the passive components. This is because clearance between the components and sockets is sometimes small and difficulty may otherwise be encountered with bulky sockets.

The solder-resist coating will minimise the effects of solder splashes etc., but it would be prudent to insert the board into the subrack from time to time during assembly, to check that the system still works.

Take care with the orientation of SIL resistor packs R17-20. The "common" connection is marked with a dot on both the resistors and the legend.

Note that IC27-31 are the spare positions.

Setting up

It has been possible to set up all the boards built so far quite accurately, without test equipment.

Turn trimpots R3 and R4 anticlockwise 24 turns to ensure that the end of the track is reached. Advance R4 by 9 complete turns. This should set the write precompensation to about 200ns (clockwise to decrease pulse width).

R3 should be advanced by 13 complete turns to set RPW (the data separator circuit) to roughly ¼ of the read clock, 500ns for 8" single density.

Rotate C5 until the fixed and moving vanes make an angle of 90°.

If an oscilloscope is available, proceed as follows:

- 1 Set drive Ø to 8", and TEST switch to "off".
- 2 Insert board, press reset and set TEST switch to "on".
- 3 Observe pulse width on pin 31 of the 2793, and adjust R4 (WPW) for the desired value of write precompensation. 200ns is suggested.
- 4 Move probe to pin 29 and adjust R3 for a pulse width of 500ns.

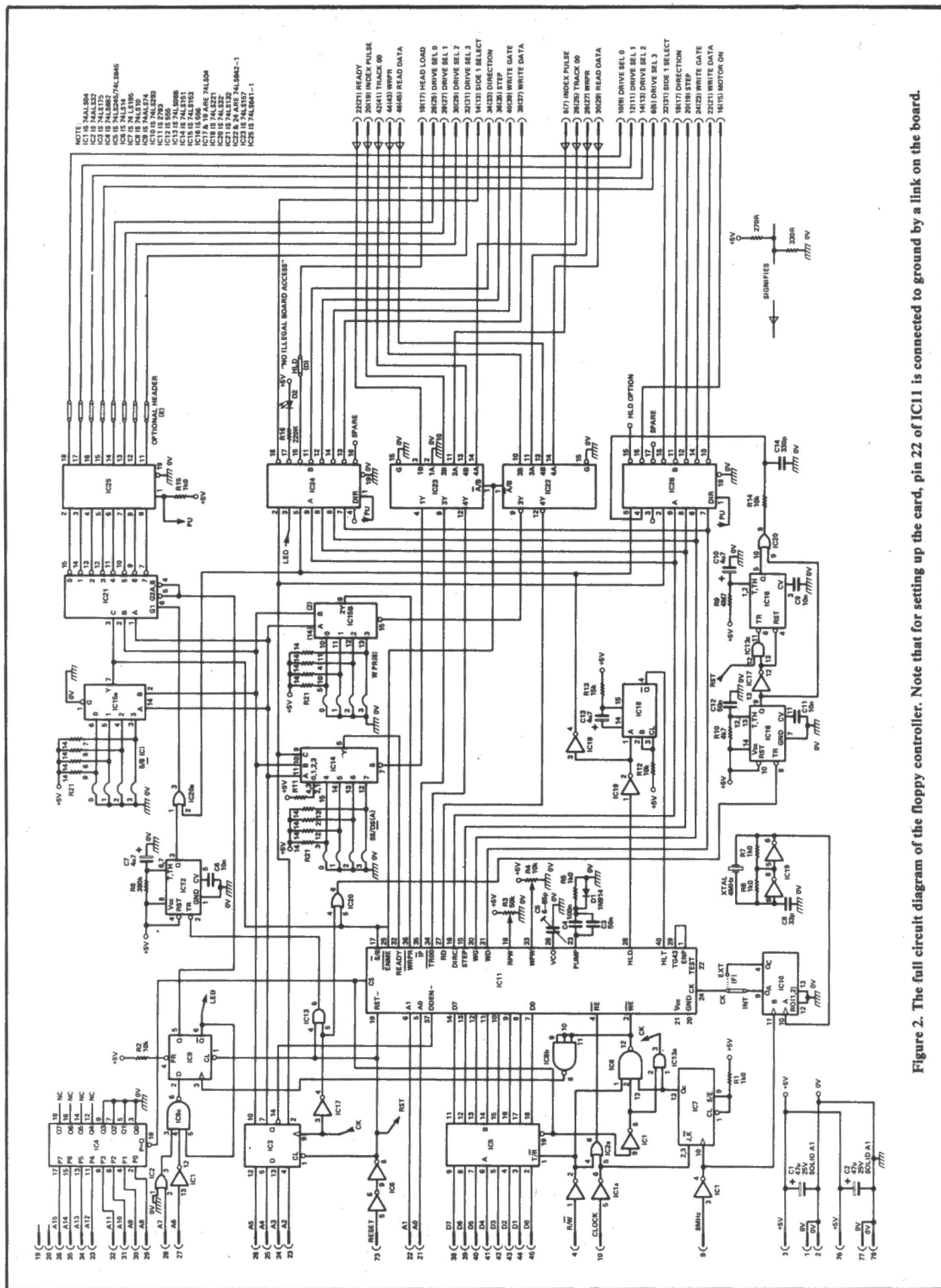


Figure 2. The full circuit diagram of the floppy controller. Note that for setting up the card, pin 22 of IC11 is connected to ground by a link on the board.

CIRCUIT DESCRIPTION

System bus interface

The 2793 requires 2 address lines to access its internal registers. The philosophy chosen for this board was to repeat those addresses for single and double data density, then again for side 0 and side 1, and finally to have a complete set of addresses for each drive. Therefore the board has a total of 64 active addresses, from \$F080 to \$F0BF. This saves having a configuration register which must be updated every time a parameter is changed.

To avoid erroneous write accesses to a disc, the board incorporates a guard band above and below the active controller addresses. A write to either guard band will inhibit the drive select logic and the system must be reset. (A LED will go out to tell you what's happened).

The board appears in the memory map from \$F000 to \$F0FF. See Fig 1 for details.

IC4 is an 8 bit comparator which detects any address in the range \$F000 to \$F0FF. Its output selects the 2793, enables both the bidirectional data buffer IC5 and write pulse gate IC8A and also enables the drive-select monostable trigger input, via IC13B. If a write occurs to an address between \$F000 and \$F0FF but not in the range \$F080 - \$F0BF, the output of IC8C(3/P/NAND) will be low. The write pulse will transfer this low to the Q output of IC9, thus disabling IC21 drive select decoder, and turning off the LED. IC9 can be cleared only by reset.

IC3 latches address bits A5 - A2, which are subsequently decoded to determine the drive, side and density that the access refers to. It is clocked only when the board is addressed.

The remainder of the bus interface is all straightforward, with the possible exception of the timing generator IC7. This is a shift register which produces a waveform similar to CLOCK (the 6809 "E" clock) but leading it by 1/4 of a cycle. This is primarily necessary to meet the 2793 address set up and Write Enable timing when using a 2MHz processor clock, but is also ensures that the set-up time of IC3 is adequately met.

2793 clock

For this board, the 2793 requires a clock of 2MHz $\pm 1\%$ with a 50% duty cycle.

IC10 is a 4 bit counter with the first and last three stages clocked separately. If the user has an 8MHz backplane clock and wishes to use this for the 2793, the track marked CK-INT must be cut and a short link added from CK-EXT. IC10 will now divide the input clock by 4. The standard option however is to use the on-board crystal oscillator, and no board changes are then necessary. The IC10 divides the 4MHz by 2 simply to generate an even mark: space ratio.

Drive select logic

The 4 inputs of IC15A specify whether the corresponding drive is 5 1/4" or 8". These inputs are normally pulled high (default to 8"), and must be grounded using a link (C) for a 5 1/4" drive.

The drive select lines are in 2 groups of 4, 0-3 for 5 1/4" drives and 0-3 for 8" drives, and are controlled by a 3-8 line decoder, IC21. Address lines A4 and A5, latched by IC3, select the drive number whilst the output of IC15A determines whether the drive is from the 5 1/4" or 8" group. IC25 is a buffer capable of sinking the 40mA necessary to select a drive.

A drive is selected on every legal access to the board. The signal which clocks latch IC3 also triggers a monostable IC12 whose output enables the 3 to 8 line decoder for 2 seconds, via OR gate IC20A. If the access is part of a disc transaction, the 2793 head load signal HLD will keep the drive selected until the end of the session. Note that IC12 is also triggered on reset, to enable the monitor EPROM to read the boot sector on power-up or reset and to go straight into FLEX (the operating system). By keeping reset pressed for more than 2 seconds, IC12 will time-out and the system will start up with the monitor prompt "C".

Side selection and write protection

Two other sets of links (A) and (B) are used to preset whether each drive is single or double sided, and to write-protect the drives if required.

The side selected is controlled by A2. After latch IC3, this signal is buffered by IC24 for the 8" drives and IC26 for the 5 1/4" drives. It also decides whether the drive select lines A4, A5 will select inputs 0-3 or 4-7 of IC14. Inputs 0-3 correspond to side 0 of drives 0-3 respectively, and 4-7 to side 1. All the inputs are pulled high, enabling both sides of every drive. However, inputs 4-7 may be grounded, forcing side 1 of these drives to be Not Ready and therefore inaccessible.

Similarly, drives may be write protected by grounding the associated inputs of IC15B. The data select inputs of this device are connected to the drive select lines, and if its corresponding input is low, the output will also be low. This tells the 2793 that it can't write to a disc in that drive.

Motor-on timer

The two halves of IC16 are configured as a retriggerable monostable of period 30 seconds, which is triggered on reset and every access to a 5 1/4" drive. (As set up by links C). The output of this monostable controls the Motor On line to the 5 1/4" drives. This facility is normally only required for 5 1/4" drives, but may be extended to 8" drives by using the spare element in buffer IC24 and removing the 5 1/4" drive trigger qualifying signal on pin 5 of IC20.

Interface to disc drives

This interface may be divided into output signals to the drives and input signals from the drives.

The outputs are all active low, open-collector and can comfortably sink the 40mA required. This is necessary because the lines are all pulled high by typically 150R to +5V, either in the drives or at the far end of the interface cable. Please note that the buffers all have the -1 suffix and that standard parts are not suitable.

The drive select outputs are all from IC25 and are via an optional header (E). This has been provided to allow the connecting tracks to be broken and drive numbers changed using cross-links on the header, instead of burrowing inside the drives.

Pin 18 of the 50 way 8" drive connector is normally spare, but is connected to HLD (head load) on this board. Provision has been made to cut this track and connect both signals elsewhere if required (D).

The input lines are all terminated by 270R to +5V and 330R to 0V, and then multiplexed by

IC22 and IC23. The signal 5/8 from IC15A selects the appropriate inputs. This circuit was used to avoid reflections from unterminated, driven cables if both 5 1/4" and 8" drives were connected via separate cables to the same inputs. The signals Index Pulse, Track 00 and Read Data are fed directly from the multiplexers to the 2793. In order to have the Side Selection and Write Protection facilities, inputs Ready and Write Protect are connected to the strobe/enable inputs of data selectors IC14 and IC15B respectively. If a selected drive contains a write protected disc, the 3Y output of IC22 will be high, disabling IC15B and forcing the Write Protect input of the 2793 active (low). Similarly, if an 8" drive isn't Ready, IC14 will be deselected and its output low. Most 5 1/4" drives don't have a Ready line, so this input is tied low, hence always Ready, at the input of the multiplexer. If you have a 5 1/4" drive with a Ready line, then cut the track grounding pin 2 of IC23 and connect it, via a spare pair of terminating resistors, to the appropriate 34 way connector pin (currently unknown).

Figure 1 Board Addresses

Address	Drive	Side	Density	2793 Register
F000 - F07F	Lower guard band			
F080	0	0	Single	Command
F081	0	0	Single	Track
F082	0	0	Single	Sector
F083	0	0	Single	Data
F084	0	0	Double	Command
F085	0	0	Double	Track
F086	0	0	Double	Sector
F087	0	0	Double	Data
F088	0	1	Single	Command
F089	0	1	Single	Track
F08A	0	1	Single	Sector
F08B	0	1	Single	Data
F08C	0	1	Double	Command
F08D	0	1	Double	Track
F08E	0	1	Double	Sector
F08F	0	1	Double	Data
F090	1	Block as for drive 0		
F0A0	2	Block as for drive 0		
F0B0	3	Block as for drive 0		
F0C0 - F0FF	Upper guard band			

A full kit of parts for the floppy controller board is available from Stirling Microsystems. See their advertisement in this issue.

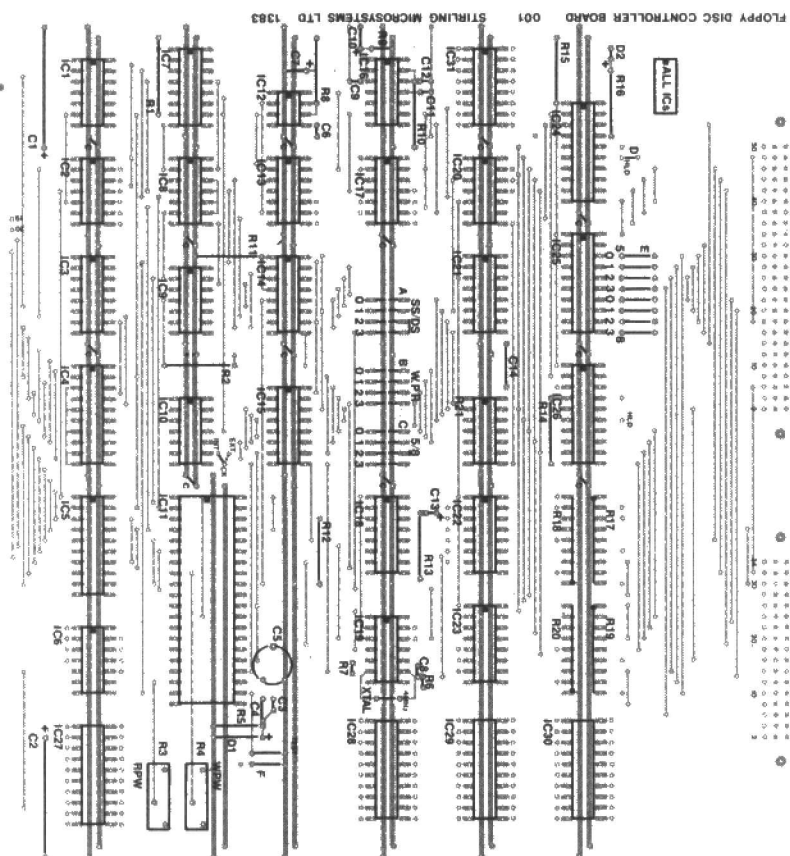
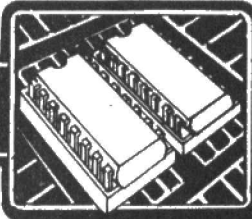


Figure 3. The overlay of the floppy card. Only the upper foil track is shown.

- 5 Monitor internal VCO frequency on pin 16. Set this to 250kHz using C5.
- 6 Return TEST to "off".

Summary of links

- A - Single/double sided. "off" or open-circuit for double-sided.
- B - Write protect. "on" or shorted to ground to write protect.
- C - Drive size. "off" for 8". (The numbers 0-3 refer to the drives).
- D - HLD connector to pin 18 of 50W connector. May be cut.
- E - Optional header for drive select lines.
- F - TEST. Used only for initial setting up.

INT-CLK - 2793 clocked from the on-board oscillator.
EXT-CLK - Option to clock 2793 from 8MHz on backplane.
(If this link is made, EXT-CLK must be cut).

E&CM

PARTS LIST

Semiconductors

IC1	74ALS04
IC2	74ALS32
IC3	74LS175
IC4	74LS682
IC5	74LS245
IC6	74LS14
IC7	74LS195
IC8	74LS10
IC9	74ALS74
IC10	74LS293
IC11	WD2793
IC12	555
IC13	74LS08
IC14	74LS151
IC15	74LS153
IC16	556
IC17	74LS04
IC18	74LS221
IC19	74LS04
IC20	74LS32
IC21	74LS138
IC22	74LS158
IC23	74LS157
IC24	74LS642-1
IC25	74LS641-1
IC26	74LS642-1
IC27-31	SPARE
D1	IN914
D2	Miniature LED

Crystal

Xtal 4MHz

Resistors

R1, 5, 6, 7, 11, 15	1k 1/4W
R2, 12, 14	10k 1/4W
R3	50k Trimpot, 3/4"
R4	10k Trimpot, 3/4"
R8	390k 1/4W
R9	4M7 1/4W
R10	4k7 1/4W
R16	220R 1/4W
R17	270R x 8 SIL
R18	330R x 8 SIL
R19	270R x 8 SIL
R20	330R x 8 SIL
R21	10k x 13 DIL

Capacitors

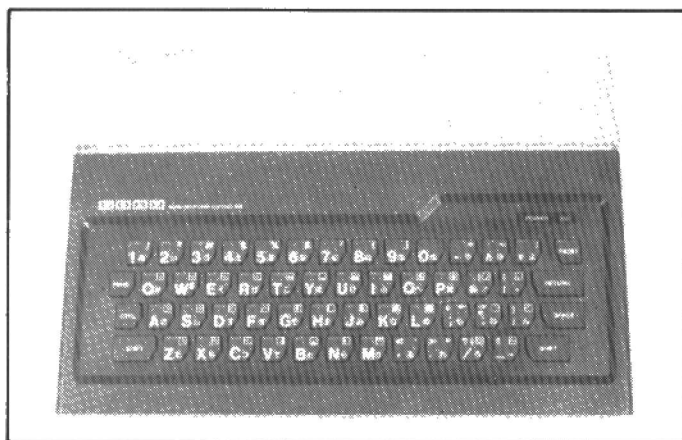
C1, C2	47u 25V Solid Aluminium
C3, C12	47n Monolithic Ceramic
C4	100n Monolithic Ceramic
C5	6-65p Trimmer
C6, C9, C11	10n Monolithic Ceramic
C7, C10, C13	4u7 35V Tantalum
C8	33p Sub. min. Plate Ceramic
C14	330p Polystyrene
C (eleven)	100n Monolithic (bypass caps)

Miscellaneous

A, B, C 4-SPST DIL Switches,
F 2-SPST DIL Switch, 34 Way & 50
Way Speedbloc PCB Mounting Plugs.

THE SORD M5

Is the Sord mightier than the pen of Gary Evans — find out with his review of a new entry to the home computer market place.



The Sord M5 is rather an enigma. On the surface the machine is a new entry into the highly competitive consumer battle ground of the £100-£200 home computer/games market. Both the M5's price and its base specification mean, however, that for a late entry to this market, the machine is saddled with less than instant appeal. Any 'skin depth' assessment of the M5 would however though, be in danger of dismissing a machine that is worthy of close attention.

Mightier Than The

In appearance, the Sord M5 is very similar to the Spectrum, the 55 keys that make up the keyboard having a similar feel to the Sinclair machine, although being fabricated of a firmer material, their action is more positive. The similarity with the Spectrum extends to the processor, a Z80 running at 3.58MHz yet the M5 is not a Spectrum clone. The differences between the two machines are very apparent upon closer inspection.

Table 1. The M5's specifications.

CPU	Z80A (3.58 MHz)
Timer	4 Channel Z80A-CTC
Memory	ROM 8k bytes, expandable up to 16k bytes by expansion cartridge. RAM 20k bytes (of which 16k bytes are used for video). Expandable up to 32k bytes in 4k-byte increments, by expansion box.
Keyboard	55 keys, 8 shifts Alphabetic: upper/lower case letters, numerics, symbols Graphics patterns 64 types BASIC statements: 28 types
Display	16 display colours, character-pattern screen (plain) 2, background screen (back plain) 1, animation screen (sprite) 32, screen modes: 4 types ● Graphic I mode: 8 x 8 dots 32 columns x 24 lines ● Text mode: 6 x 8 dots 40 columns x 24 lines ● Multi-colour mode: 4 x 4 dots 64 columns x 48 lines ● Graphic II mode: 256 x 192 dots (full-graphic)
Synthesizer Interfaces	3 chords, 1 noise, 7 special sounds Rf output terminal: RCA phono jack: NTSC Video output terminal: RCA jack: NTSC composite video output 1 Vpp 75 ohms Sound output terminal: RCA pin jack; output voltage 1 Vpp Joypad terminals: 6-pin mini-DIN x 2 (L & R) Audio cassette terminal: 8-pin DIN, with a Remote terminal Writing speed 2000 bits/s Printer terminal: Special 16-pin flat-cable connector based on the Centronics interface Cartridge terminal: internal bus 56-pin edge connector Power Supply terminal: 6-pin DIN

The M5's specifications are outlined in **Table 1** and from this it can be seen that the way in which the Sord's memory is configured is rather unusual. Followers of the *E&CM* Hi-res Computer will be familiar with the philosophy of dividing system memory between video display RAM and user RAM. The Sord uses a similar dedicated video control IC (more on this later) and separate areas of dedicated RAM. Thus the 20K of RAM featured in the M5's basic specification is configured as 16K video display RAM and 4K user RAM. By the time various housekeeping and variable storage requirements are fulfilled by the 'user RAM', there are only about 3K bytes available for user programs.

The fact that there will never be a conflict between the two areas of RAM means, that even in a hi-res graphics mode, these 3K bytes of user RAM will still be available — programs will not be 'squeezed out' as the display's memory demands increase. The small amount of user RAM will, even if the above is borne in mind, conjure up images of the ZX81's memory shortcomings and, with the market being used to machines that feature 48K of memory, this area of the M5 specification may well make many people think twice about the machine.

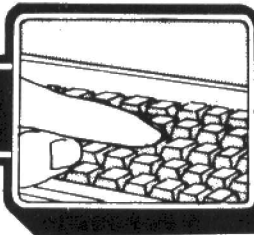
ROM In A Box

The way in which the M5's ROM is packaged is also a departure from the trends set by the machine's most likely competitors. The Sord provides 8K of ROM on board, this providing the M5's monitor and housekeeping routines, with additional ROM being added to the computer by way of its cartridge expansion slot. Up to 16K of additional ROM can be added in this way — the machine being supplied with one cartridge providing an Integer only Basic — BASIC I.

Starting Out

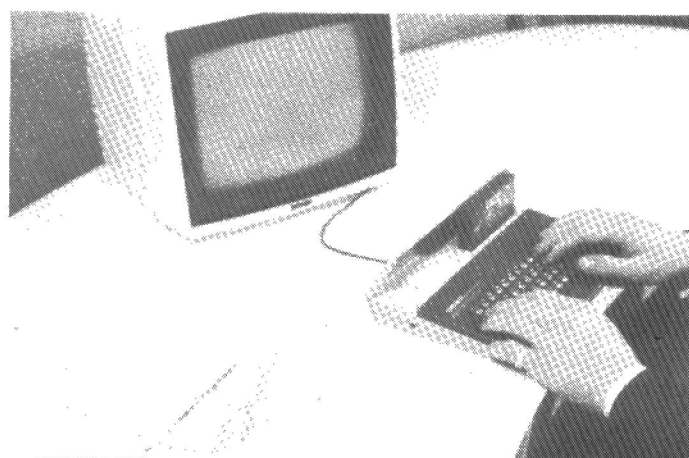
In addition to the BASIC — I cartridge the M5 is supplied with a separate power supply, two manuals and a cassette tape that features a program to aid tuning a TV to the frequency of the Sord's modulator (the usual Ch. 36) as well as two games programs.

The first of the manuals is designed to enable even a computing tyro to get the machine up and running, with large diagrams of the way in which the various components fit together being the order of the day. The power supply and cassette recorder both plug into the computer via DIN connectors but it's nice to note that the connectors are, respectively, of the six and eight pin variety and thus there is no possibility of damage to the M5 by confusion as to which is which. Confusion does arise, however, when it comes to connecting the three jack plugs at the other end of the cassette lead to a recorder. Although each plug is colour coded, the manual makes no mention of which plug is which. No damage



can be done to either computer or recorder if the plugs are used in the wrong order and, as the 2.5mm plug is definitely for controlling the remote option, it's only a choice between the two 3.5mm plugs that has to be made. Of these logically(?) the red plug is for recording a program and should be connected to the recorder's mic socket.

Video output from the computer is either via the RF section of the TV to be used as the monitor or from the M5's video output socket that supplies a standard composite video output.



BASIC - G

Although the M5's specification proclaims that the machine is capable of sophisticated graphic and sound generation, the full power of these facilities cannot be realised without the BASIC - G cartridge. For example, the BASIC - I manual makes no reference to sound generation, and although we managed to get the M5 to squeak and squeal by using the OUT command, any music generation was out of the question (excuse the pun).

With BASIC - G resident the full power of the M5's Texas video controller is realised. This IC uses a sprite based system, with graphics being displayed on a number of sprites (planes) numbered from 0 to 31. There is no room here for a lengthy explanation of sprites 'but briefly' a sprite is a dot matrix (varying in size from 8 x 8 to 16 x 16). Sprites can be assigned to one of 32 planes with various BASIC - G commands controlling the attributes and movements of a sprite on any specific plane.

An interesting aspect of the sprite system is that a sprite on level 0 will always occlude sprites on planes with a higher number. Thus by careful selection of sprite level when building up a graphic display, the relationship between the movement of foreground and background objects can easily be defined.

Table 2. The commands available with BASIC-I.

Direct Commands	Screen Control Commands
AUTO	LOC
CLEAR	MAG
CLS	SCOD
CONT	SCOL
	STCHR
DEL	
LIST	VIEW
LIST #2	VPOKE
NEW	
RUN	
	Functions
	ASCII
	CHR\$
	HEX\$
	INKEY\$
	LEFT\$
	LEN
	MID\$
	RIGHT\$
	VAL
	Operating Commands
	CURSOR
	ERR
	ERRL
	ERRLS
	PEEK
	VPEEK
	Value Functions
	ABS
	FRE
	INP
	NUM\$
	RND
	SGN
	TIME
Input/Output Commands	
CHAIN	
DATA	
INPUT	
OLD	
OUT	
PRINT	
PRINT #2	
READ	
RESTORE	
SAVE	
SAVE	
TAPE	
VERIFY	
Program Commands	
CALL	
DIM	
END	
FOR-TO-STEP	
GOSUB	
GOTO	
IF-THEN-ELSE	
LET	
NEXT	
POKE	
RANDOMIZE	
REM	
RETURN	
STOP	

BASIC I

The M5 will not function without a cartridge in position and thus before switching power on the BASIC - I cartridge should be slotted into the computer. The cartridge is pushed into a slot on the top surface of the machine and fits snugly with no possibility of any 'wobble'.

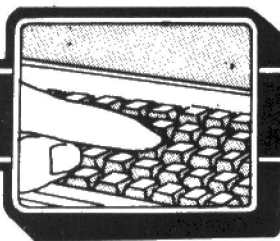
The separate power switch provides an on/off switch that is most welcome as it means its not necessary to grope around for the mains plug to reset the M5 if it ever locks up.

With BASIC - I up and running the computer provides a fairly standard basic package, if a rather limited one, coping only with integer arithmetic. The FUNC key to the upper left of the keyboard allows single key entry of many basic keywords (again reminiscent of Sinclair basic) although all of these can be entered in full. Although the keywords are entered 'at a stroke' they do not appear to be stored as tokens but rather as their equivalent ASCII strings - thus the M5 does not differentiate between commands entered with the aid of the FUNC key or those typed in full.

The CNTL key, just below the FUNC key, in conjunction with various other keys, allows full control of the cursor for editing text and placing the M5 into its various display modes.

Although integer only, BASIC - I does provide a number of 'up-market' functions. AUTO for example allows automatic line numbering and a number of commands associated with the graphics IC provide powerful, if limited control of graphics blocks. A full list of BASIC - I functions appears in Table 2.

Entering programs was made easy by the nice feel to the keyboard and the fact that each key press was acknowledged by a 'bleep' reproduced via the TV set's speaker.



Sound Generation

An additional dedicated IC endows the M5 with an easy to use and sophisticated sound capability. A typical command would be

PLAY "EGAGEGAG", "CBA"

The PLAY command, followed by the notes designation is all that is required to generate music and enclosing two sequences of notes in the same PLAY statement generates harmony – up to three notes can be generated simultaneously. The M5 also allows for varying the volume level of the notes produced and to alter the rhythm of the music generated.

The addition of BASIC – G also adds the RENUMBER command to the BASIC as well as allowing the joystick ports to be read. These will return a value from 0 to 8 depending on the direction of the joy pads control.

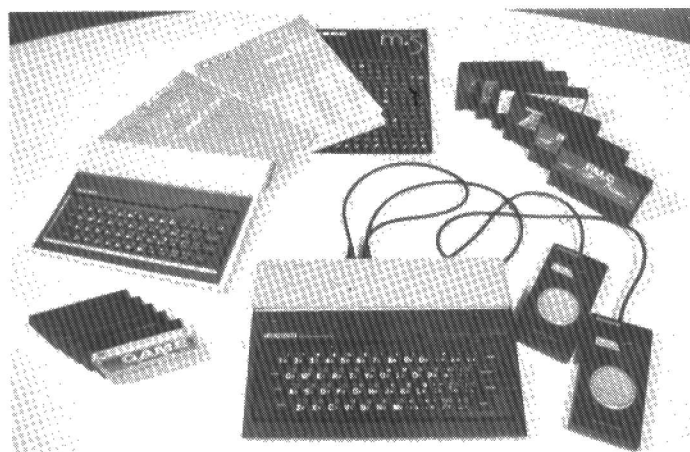
FALC

Another cartridge available for the M5 is FALC. This is a versatile data handling package that allows a wide range of information to be entered and manipulated. Again, there's no room here to go into a detailed look at the package, it deserves a review in its own right, but it is an easy to master and powerful data management system.

The addition of the FALC cartridge also provides a floating point capability.

Interfaces

In addition to the joystick interfaces, the M5 features a centronics type printer port. **Table 3** shows the connections and it should be possible, with perhaps a fudging inverter or two, to drive any parallel printer directly from the M5.



Rounding Off

The Sord M5 is produced to a very high standard. A look inside reveals well designed boards with plenty of shielding in evidence to obviate RF interference. No 'bodge' boards visible either, a sign that the Sord is not a machine that has been rushed onto the market. It's unlikely therefore that any nasty hardware or software bugs are likely to rear their heads.

At £190 the Sord is an expensive machine and for this price, the fact that only 4K of user RAM is provided and that the basic is integer only, must be seen as a minus. Adding BASIC – G (which at £34.95 pushes the price beyond the £200 level), does however provide an excellent games machine.

Table 3. The Sord M5's printer port pin assignment.

Pin 1	$\overline{\text{PSTB}}$	Pin 2	GND
Pin 3	PD1	Pin 4	PD2
Pin 5	PD3	Pin 6	PD4
Pin 7	PD5	Pin 8	PD6
Pin 9	PD7	Pin 10	PD8
Pin 11	—	Pin 12	PBUSY
Pin 13	GND	Pin 14	GND
Pin 15	PCOM	Pin 16	$\overline{\text{PSTB}}$

Adding BASIC – F, the floating point package, again at £34.95, will turn the M5 into a powerful maths/scientific computer while FALC turns the M5 into a tidy information management machine.

A series of games cartridges are also available – these include Tank Battalion and Word Maze, while the cassettes for the M5 each feature two games, a few from the range of seven at present available being Solitaire/Tower of Hanoi and Black Jack/Slot Machine.

The one M5 add-on not yet mentioned as regard to price is the pair of Joysticks – these weigh in at £24.95 the pair.

Last Word

The M5 is being marketed very much on its quality and reliability, and it certainly lives up to its promise in these areas at least as far as the M5 itself goes. The manual provided with the Basic-I cartridge leaves a lot to be desired with many of the functions receiving scant, if any mention. A supplement to the manual is being prepared however and should be supplied with the M5s that are sold over the counter.

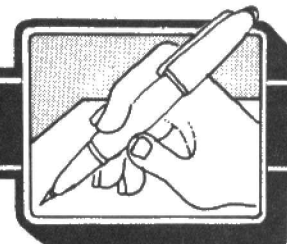
The M5 was well liked, and if potential owners accept that quality must be paid for, it should prove a good buy. In Japan 17,000 M5s were sold in the first three months of this year, with production running at 23,000 units. The failure rate of M5s in the field was quote 'very low indeed'.

Post Script

The M5 is also to appear under the banner of Computer Games Limited. CGL's name may not be too familiar, but the company have been involved in the distribution of Computer Games from the early days of the Fidelity Chess Computer. The M5 moves them up-market from their 'game & watch' distribution set-up, but their established links with many of the major High Street outlets will mean that the M5 will have a very high profile from 'day one'.

Even with this degree of exposure though, the M5 will probably struggle to achieve volume sales without a significant drop in price.

E&CM



Spanish ZX Club

We're the best ZX User's Club in Spain. We edit monthly a newsletter 30-40 pages each.

We teach Free basic courses each week.

We meet every day from 18 to 19.30 p.m. We do make free interchange of programs. We even have our own Pet "ZX Captain". To contact us, write to:-

Mr. C. Benito,
P.O. Box 3253,
Madrid,
Spain.

Dear Sir,

The fact that ZX computers are spreading like wildfire can only be explained by the rather too seductive Sinclair Basic. Any other program, consisting to a large extent just of 'noughts and crosses', is suitable for experts only.

The Cambridge boys gave Sinclair and the world a simple method of communication between machines and ordinary homo sapiens.

K. A. Meyer
Sidney
Australia

Dear Sir/Madam,

In the April (1983) edition of your magazine (page 62) there was an article to convert a colour television set into a colour monitor, using a BBC (model B) micro. Please could you tell me if it is possible to do the same conversion to a Triumph CTV 8200 model, as the conversion shown is for a Ferguson TV.

I have enclosed a circuit diagram of my TV, so if any alterations are needed you could let me know.

Yours sincerely,
Matthew Hawder
Sheffield
S. Yorks.

We have had a lot of letters along the lines of yours, Mr. Hawder, and unfortunately cannot be of much help. The conversion relies on the fact that the TV set to be operated on features the TDA3560 video processing IC. If the circuit diagram or a TV set reveals that this IC is indeed at the

heart of the video circuitry, it should be possible to go ahead with the conversion as outlined in the article. If this is not the case, very different techniques would have to be employed and we are not in a position to supply such information.

Dear Sir,

I am dismayed that on many occasions I buy a copy of your excellent magazine only to find that an article of particular interest to myself has been split into two or more parts and that I must wait an exasperating four weeks before the next installment appears. Please, please publish all the details of an article in just one part.

Noel Holban
London SW

We'd like to be able to do just that Mr. Holban, but we just do not have enough editorial pages available to do so. In order to publish some of our projects in one issue might eat up 10% - 20% of our pages. Adopting that sort of strategy would seriously disturb the variety and range of items published in an issue and it's this variety that we're keen to maintain.

If any other of our readers have any comments on this matter or on the advisability or not of publishing page after page of software listings, please write and let us know.

Dear Sir,

In your 'Comment' article of the May issue of *E&CM* you made no apologies for NOT being a first time users magazine. Fine - I've no objection to playing with the big boys; but then you have the brass neck to publish a review of Dragon software that consists entirely of games!

Are there no wordprocessors? No spreadsheets (are there?). No databases? How about an article on using Compusense's DASM & DEMON? Are there no brave publishers willing to take use ingenuities by the hand and sell us the sacred books that will initiate us into the mysteries of Assembler and Machine Code? Is Dragon Forth worth a fig? (Sorry!). I think you get my drift.

Would you also consider an article on basic techniques such as soldering/desoldering etc? What sizes of iron to use? What flux? How does one desolder and reconnect one tiny item on a PCB without melting everything around it? How about the elements of making up a PCB?

A review of basics from time to time, even for the elite, can do no harm and not a little good. Is there a book on the subject you could recommend?

Strength to your elbow.

Yours faithfully,

D. L. George
Shoreham-by-Sea
West Sussex

Yes, we get your drift Mr. George, perhaps this month's Wordwise and View reviews are more up your street. We'd also planned an excellent article on the BBC's assembler but you'll have to wait until next month for that as we ran out of pages this month.

On the subject of basic constructional techniques, watch out for a new series starting soon.

House Style

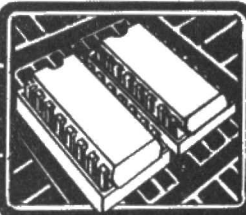
Some magazines feature regular half page appeals for authors. We're not about to start a similar advertising campaign as we have a regular flow of quality articles arriving at our offices - that's not to say we don't want you to keep them coming though.

We do get a number of requests for guidance on the style of articles though, so here are a few words on the subject.

Articles should, in general, be about 2000/3000 words in length and be typed; double spaced, with wide margins. All diagrams should be on separate sheets of paper and should be as clear as possible. If appropriate, photographs should be enclosed, each having a brief description attached to them.

Any material submitted should not contain any original artwork and you should keep a copy of your manuscript.

We aim to acknowledge all articles submitted by return and to assess them within a period of four to six weeks.



Spectrum Image Processor

Part 2

Richard Sargent and Robert Harvey describe a higher resolution version of their image processing system described last month.

Resolution v. Brightness Levels

The first display program used the Spectrum Attributes and gave us a reasonable spread of brightness levels, from 0 to 7 but this was quite fortuitous. The reason the Attributes were used at all was for simplicity of coding. They cannot be printed out because they represent paper colour, and they forced upon upon us the acceptance of extremely coarse resolution, 24 x 32. Each Attribute "building block" also consists of 64 pixels, but these were not being illuminated. Let us forget the Attributes and concentrate upon the pixels. There are 256 across the screen and 192 down the screen. Each can either be illuminated or not illuminated. If we are prepared to accept just these two brightness levels, on and off, then we can have a picture resolution of 256 x 192. This is one extreme. Now consider the building block of 64 pixels. This is a character square, eight by eight. We could arrange the pixels within a character square to give 64 different shades of brightness by arranging 64 different types of character square, each with a dot density changing in increments of one from zero to sixty-three. But we would be back to a resolution of 24 x 32. Clearly there is a trade-off between resolution and levels of brightness and it is possible to reach a happy medium. The Spectrum character square is 8 bits/1 byte wide. If we construct a square 4 bits/1 nibble wide then we achieve a resolution of 48 x 64 and each square can take on one of 16 levels of brightness because it contains 16 pixels. Working in "nibbles" will present no problems at machine-code level. Fig. 1 shows the 16 "Mosaic" patterns chosen to represent the brightness levels.

The Software

The core program is essentially unchanged. As with the first version we

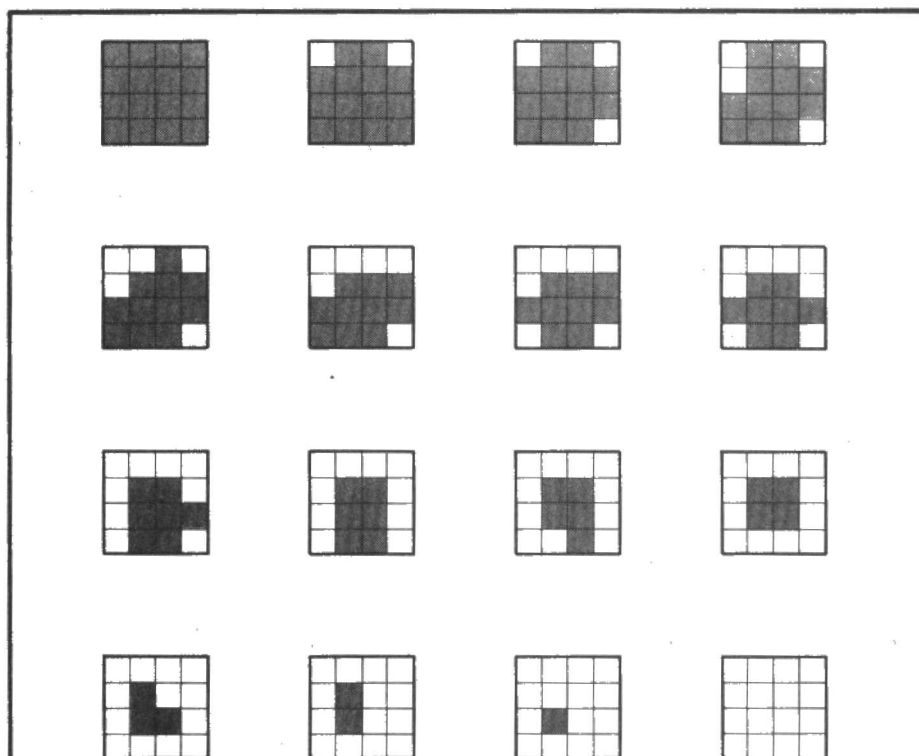


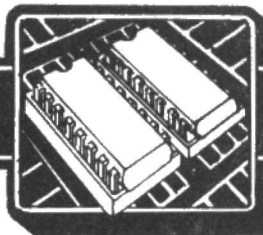
Figure 1. The mosaic patterns held in a table at 7176H.

only capture as much data as we intend to screen, in this case 3072 bytes. Label LINES must change to 48 and label CTR\$ST changes to 9FH. These are the vertical and horizontal counters and must be changed for the new resolution. The calculation of CTR\$ST may look complicated: It is (Horizontal samples ★ Displacement) + (Finish of scan-1); but if it is built up in this manner it is an easy matter to experiment with different values of displacement, the interesting effects of which we shall see later.

New Display Routine

As with the previous Display routine, the brightness values for all the picture points are in RAM starting at location "STORE". This time there are 600H

bytes waiting to be processed. Each byte is holding a positive number in the range 0 - FFH, but because of the hardwiring bits 7 & 6 do not represent brightness data and must later be rejected. There are only a maximum of 64 different values present, with bit 7 of the ADC's output residing in bit 5 of the memory byte and so on down to bit 2 of the ADC's output which is in bit 0 of memory. Bits 0 and 1 of the ADC's output were never recovered from the ADC interface circuit. An AND 3FH masks out the unwanted bits 7 & 6 and we are left with a brightness value in the range 0 - 64 in the Accumulator Register. This number is used as a pointer to get into the 64 byte LOOKUP table.



LOOKUP a brightness level

In the software example provided there is a linear relationship between the brightness value 0 – 64 from the ADC and the user-provided brightness levels 0 – 15 held in the LOOKUP table. This relationship can easily be altered. The middle of the LOOKUP table holds mid-tone values, and these may be replaced by bright values or dark values, thus affecting the contrast of the displayed picture. A simple BASIC program could arrange for the LOOKUP table to be changed in a sequential and predetermined way thus sweeping a tone change across the displayed picture. A slow random change imposed upon the LOOKUP table will see the displayed picture turn gradually into a pattern of square blocks.

However, any value entered into the LOOKUP table must be in the range 0 – 15. The software uses this value as a pointer into the MOSAIC table where the correct pixel mosaic relating to the 16 brightness values is obtained.

Mosaic Pattern

The Spectrum screen as we have seen is $256 \times 192 = 49152$ pixels in size. It is bit-mapped which means that one bit of any screen-memory represents one pixel on the screen itself. It follows that $49152/8 = 6144$ bytes are required to hold all the screen information. A character square contains 64 pixels and requires 8 bytes of screen-memory storage. The addresses of these 8 bytes do NOT run consecutively, as every Spectrum owner must now surely know, and many hints have been published on how to reference an 8 byte character square. Fig. 2 shows the addresses for the 8 bytes of the first character square (top lefthand corner of the screen, Row and Column co-ordinates 0,0). This character square is displaying the mid-tone mosaic (DEFB 0,6,7,0) in its top-left quadrant.

This arrangement is not as bad as it may first appear, since the addresses are advancing down the character square in jumps of 100H, 256 decimal. If the address is contained in register HL then the instruction INC H will step on to the next address. To make matters a little more complicated a new starting address must be computed for the beginning of every ROW and this in turn will depend

upon which ZONE the ROW happens to be in. The Spectrum screen is divided into three equal, horizontal zones, each being 2K bytes long. The start of zone 1 is, as we have seen, 4000H. The start of zone 2 is Row 8, Column 0, 4800H. The start of zone 3 is Row 16, Column 0 5000H. The last byte of zone 3 is address 57FFH.

ROW-COLUMN to ADDRESS conversion

To maintain sanity the program "thinks" in terms of Row-Column co-ordinates and uses one short subroutine (CONVERT) to calculate the required display-file addresses and this routine is used by both the display routine and the printer routine.

Address R=0 C=0	Content	Screen pattern	Address R=0 C=1
4000H	0 0H	0 0 0 0 0 0 0 0	4001H
4100H	6 0H	0 1 1 0 0 0 0 0	4002H
4200H	7 0H	0 1 1 1 0 0 0 0	4101H
4300H	6 0H	0 1 1 0 0 0 0 0	etc
4400H	0 0H	0 0 0 0 0 0 0 0	
4500H	9 0H	0 0 0 0 0 0 0 0	
4600H	0 0H	0 0 0 0 0 0 0 0	
4700H	0 0H	0 0 0 0 0 0 0 0	
Address R=1 C=0			
4820H			
4120H			
etc			

Figure 2.

BUILDING THE PATTERN ON THE SCREEN

The correct pattern of pixels, extracted from the MOSAIC table is always 4 bits wide (a "nibble"). The left-hand side of the 8 bit character square is loaded with a mosaic nibble and later another mosaic nibble is loaded into the right-hand side of the character square in a manner which preserves the left-hand side. The instruction is OR (HL), where HL is pointing at screen memory. The

accumulator is holding a valid low nibble and is ORed with the contents of memory pointed at by HL, which of course already holds a valid high nibble. Thus two patterns are loaded into the top half of each character square, and we have a resolution of 64 across the screen. Down screen the resolution is 48.

OUTPUT TO AN MX80 PRINTER

The software for the printer routine is specific to the MX80, but the principle of operation remains the same for most of the common dot-matrix bit-addressable printers. In "bit-image" mode the printer acts as though it has 8 wires in its print-head, and scans along the paper building up a high-resolution image which is 8 dots deep for any one graphic line. The line itself can be of a specified dot-count in length, up to the maximum number of dots that the printer can manage horizontally. The printer expects a full line's worth of data to be transmitted to it, terminated by 0DH (carriage return), whereupon it will print that line out. For the Epson 80 the expected format is shown in Fig. 3.

We can extract the data to be printed from the display file of the Spectrum, but it is arranged in a horizontal sequence of bits within bytes since this is the pattern that any TV monitor expects: the TV raster expects to build up the TV picture one dot-high line at a time. The printer, on the other hand, wishes to build up a line which is 8 dots-high, and expects to find the bits correctly ordered to achieve this.

Rotate And Be Merry

The program steps through the display file in a logical order determined by Row and Column co-ordinates, which are

1BH, 4BH, LL, HH, ddddddd, ..., 0DH

1B4B means "accept LLHH as a binary number representing the length of graphic data" Changing 4B to 4C will cause printing to be done in "double density" mode: the horizontal axis is compressed to 50% of the original.

LLHH is the length in standard lsb-first binary notation.

dddd.. are the bytes of graphic data. In each byte the most significant bit represents the uppermost dot of the vertical eight to be printed.

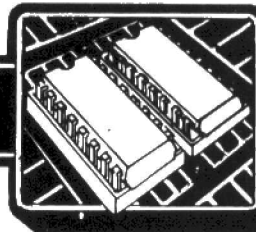
Figure 3. Data format expected by the Epson 80 printer.

391C

```

7801 DD21B671 DISPLAY LD IX,STORE ;Counts the horizontal nibbles
7805 0600 LD C,0 ;Counts the vertical nibbles
7807 0600 HORIZLP LD B,0
7807
7809 DD7E00 VERTLP LD A,(IX+0) ;Obtain brightness
780C 2F CPL ;Software invert
780D E63F AND 3FH ;Mask out bits 6 & 7
780D ;Brightness ;Now in the range
780D ;conversion ;0 - 64
780F C5 PUSH BC ;but this program
780E 5F LD E,A ;requires a range
78E1 1600 LD D,0 ;of 0 - 15 so
78E3 213671 LD HL,LOOKUP ;begin brightness translation
78E4 15 ADD HL,DE ;by means of LOOKUP table
78E7 9E LD E,(HL) ;New value in E
78E7 ;DE is now an offset value
78EB 217671 LD HL,MOSAIC ;Point to start of pattern table
78EC EB EX DE,HL ;Set offset into HL and
78EC 29 ADD HL,HL ;multiply
78EE 19 ADD HL,HL ;by four
78EF ED ADD HL,DE ;so pointing HL
78EF EB EX DE,HL ;and now DE at the first byte
78EF ;of a group of four "pattern bytes"
78F0 CD1671 CALL CONVERT
78F0
78F0 ;DE & HL valid so put
78F0 ;mosaic onto screen
78F3 0604 LD B,4
78F5 1A LD A,(DE) ;load up a pattern byte
78F6 CB41 BIT 0,C ;Odd or even nibble ?
78F8 2806 JR NZ,PAT1
78FA 3600 LD (HL),0
78FC 07 RLCA
78FD 07 RLCA
78FE 07 RLCA
78FF 07 RLCA
7900 B6 PAT1 OR (HL) ;One quarter of the mosaic
7901 77 LD (HL),A ;goes onto the screen
7902 13 INC DE ;Advance DE to grab the next quarter
7903 24 DROP ;Drop pointer down by adding 256
7904 10EF DJNZ DLOOP ;and go round to complete mosaic
7906 C1 POP BC
7906
7907 DD23 INC IX ;Set next stored byte
7909 04 INC B ;Set next vertical nibble
790A 70 LD A,B
790B FE30 JC LINES
790D 3BCA JR C,VERTLP ;if 40 nibbles done, fall through
790D
790F 0C INC C ;Set next horizontal nibble
7910 79 LD A,C
7911 FE40 CP NCNT
7913 3BC2 JR C,HORIZLP ;if all 64 nibbles done
7915 C9 RET ;picture mosaic complete
7915
7915 ;address conversion
7916 79 CONVERT LD A,C ;Use B & C software counters
7917 CB3F SRL A ;to create a Spectrum
7919 6F LD L,A ;display-file address in
791A 70 LD A,B ;register HL
791B E630 AND 3FH ;00110000
791D 0F RRCA
791E 67 LD H,A
791F 70 LD A,B
7920 E60E AND 0EH ;00001110
7921 07 RLCA
7922 07 RLCA
7923 07 RLCA
7924 07 RLCA
7925 07 RLCA
7926 B5 OR L
7927 6F LD L,A
7928 70 LD A,B
7929 E601 AND 1 ;00000001
792B 07 RLCA
792C 07 RLCA
792D B4 OR H
792E F640 OR 40H
7930 67 LD H,A
7931 C9 RET
7931
7932 00000000 DB 0,0,0,0
7932
7932 ;-----
7932
7933 00000000
7934 01010101 LOOKUP DB 0,0,0,0,1,1,1,1
7935 02020202 LD L,A
7936 03030303 DB 2,2,2,2,3,3,3,3
7937 04040404 LD A,B
7938 05050505 DB 4,4,4,4,5,5,5,5
7939 06060606 RLCA
7940 07070707 DB 6,6,6,6,7,7,7,7
7941 08080808 RLCA
7942 09090909 DB 8,8,8,8,9,9,9,9
7943 0A0A0A0A LD B,10,10,10,10,11,11,11,11
7944 0B0B0B0B LD L,A
7945 0C0C0C0C DB 12,12,12,12,13,13,13,13
7946 0D0D0D0D RLCA
7947 0E0E0E0E DB 14,14,14,14,15,15,15,15
7948 0F0F0F0F
7949 00000000 DB 0FH,0FH,0FH,0FH,0,0,0FH,0FH
794A 01010101 DB 0,0FH,0FH,0EH,0,7,0FH,0EH
794B 02020202 DB 2,7,0FH,0EH,0,7,0FH,0EH
794C 03030303 DB 0,7,0FH,0,0,0,0FH,0
794D 04040404 DB 0,0,0,2,0,0,0,0
794E 05050505 DB 0,4,0,0,0,4,0,0
794F 06060606 DB 0,0,4,0,0,0,0,0
7950 07070707 ;-----
7951 08080808
7952 09090909 STORE DS NCNT*BYTES/2 ;600H long
7953 0A0A0A0A
7954 0B0B0B0B FIN EQU %
7955 0C0C0C0C
7956 0D0D0D0D
7957 0E0E0E0E
7958 0F0F0F0E

```



Basic loader routines.

```

1000 CLEAR 28415
1010 LET A = 28548
1020 LET A = A + 1
1030 READ D : POKE A,D
1040 IF A < 28586 THEN GOTO 1020
1042 PRINT "Start cassette"
1044 PAUSE 0
1050 SAVE "printer" CODE 28549,37
1060 STOP
1070 REM PRINTER DRIVER ROUTINE
1072 DATA
197, 245, 62, 207, 1, 31, 2, 237, 121, 62, 255, 237,
121, 62, 15, 1, 31, 3, 237, 121, 1, 31, 0, 237,
120, 203, 71, 32, 250, 241, 1, 31, 1, 237, 121, 193,
201

```

```

T 6F85 6FAA 10 8 0011
6F85 C5 F5 3E CF 01 1F 02 ED 79 3E FF ED 79 3E 0F 01
6F95 1F 03 ED 79 01 1F 00 ED 78 CB 47 20 FA F1 01 1F
6FAS 01 ED 79 C1 C9 FF 00 FF 00 FF 00 FF 00 FF

```

```

1000 CLEAR 28415
1010 LET A = 28800
1020 LET A = A + 1
1030 READ D : POKE A,D
1040 IF A < 29110 THEN GOTO 1020
1042 PRINT "Start cassette"
1044 PAUSE 0
1050 SAVE "mosaic" CODE 28801,229
1060 STOP
1070 REM MOSAIC ROUTINE
1072 DATA
221, 33, 0, 88, 33, 245, 112, 17, 32, 0, 14, 32,
6, 24, 221, 229, 126, 47, 230, 56, 221, 119, 0, 221,
25, 35, 16, 244, 221, 225, 221, 35, 13, 32, 233, 201,
26, 203, 65, 32, 6, 54, 0, 7, 7, 7, 182,
119, 19, 36, 16, 239, 193, 221, 35, 4, 120, 254, 48,

```

```

56, 202, 12, 121, 254, 64, 56, 194, 201, 121, 203, 63,
111, 120, 230, 48, 15, 103, 120, 230, 14, 7, 7, 7,
7, 181, 111, 120, 230, 1, 7, 7, 180, 246, 64, 103,
201, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,
1, 2, 2, 2, 2, 3, 3, 3, 3, 4, 4, 4, 4,
4, 5, 5, 5, 5, 6, 6, 6, 6, 7, 7, 7, 7,
7, 8, 8, 8, 8, 9, 9, 9, 9, 10, 10, 10, 10,
10, 11, 11, 11, 11, 12, 12, 12, 12, 13, 13, 13, 13,
13, 14, 14, 14, 14, 15, 15, 15, 15, 15, 15, 15, 15,
15, 6, 15, 15, 15, 6, 15, 15, 14, 6, 7, 15,
14, 2, 7, 15, 14, 0, 7, 15, 14, 0, 7, 15,
6, 0, 6, 15, 6, 0, 6, 7, 6, 0, 6, 6,
6, 0, 6, 6, 2, 0, 6, 6, 0, 0, 4, 6,
0, 0, 4, 4, 0, 0, 0, 4, 0, 0, 0, 0,
0

```

```

1000 CLEAR 28415
1010 LET A = 28415
1020 LET A = A + 1
1030 READ D : POKE A,D
1040 IF A < 28548 THEN GOTO 1020
1042 PRINT "Start cassette"
1044 PAUSE 0
1050 SAVE "print" CODE 28416,132
1060 STOP
1070 REM SCREEN DUMP CODE
1072 DATA
62, 27, 205, 130, 111, 62, 65, 205, 130, 111, 62, 8,
205, 130, 111, 62, 13, 205, 130, 111, 62, 10, 205, 130,
111, 6, 0, 14, 0, 221, 33, 0, 91, 205, 22, 113,
197, 6, 8, 229, 22, 0, 62, 8, 78, 88, 203, 37,
29, 32, 251, 203, 10, 36, 61, 32, 243, 122, 221, 119,
0, 221, 35, 225, 16, 229, 193, 12, 12, 121, 254, 64,
56, 215, 205, 85, 111, 4, 120, 254, 48, 56, 199,
201, 197, 62, 27, 205, 130, 111, 62, 75, 205, 130, 111,
62, 0, 205, 130, 111, 62, 1, 205, 130, 111, 33, 0,
91, 6, 0, 126, 205, 130, 111, 35, 16, 249, 62, 13,
205, 130, 111, 62, 10, 205, 130, 111, 193, 201, 195, 133,
111

```

loaded into the B and C registers. The Row and Column method is used so that we can use the CONVERT subroutine to collect accurate display file addresses into the HL register. Meanwhile the IX register is pointed at a spare area of RAM called BUFFER. The 64 bits from the eight widely-separated bytes of a display file character square are captured, rotated, and placed sequentially into BUFFER. (HL) is the source, (IX) the destination. Rotation is carried out in the C register, the amount of rotation needed being dependent on the state of the B register (Row count). Results of rotations are built up in the D register. A scan of 32 columns (when C=4) will have built up 256 bytes in the BUFFER. These bytes are then sent to the printer by the routine OUTBUFF.

A BASIC loader for the printer routine is provided, and the screen dump command is RANDOMISE USR_____.

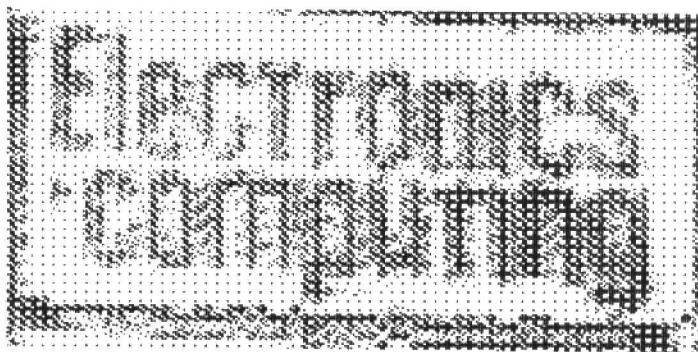
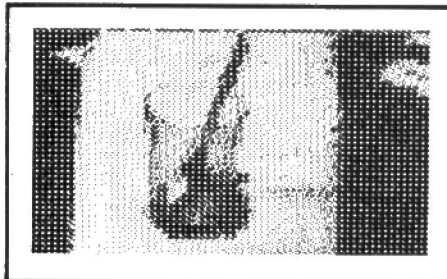
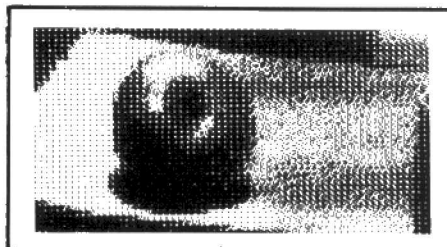
Conclusion

The techniques of processing video pictures offers something of a challenge to software writers. Computers which offer individual pixel colours like the BBC micro or the LYNX will provide all the shades of tone needed at pixel-resolution level, resulting in picture resolution approaching that of a standard TV picture.

Printer resolution may present a problem to which software must provide a solution. A square printed by an Epson 80 is 6 dots wide and 8 dots deep: to get the correct aspect ratio of 1:1 either the sampling software or the printing software could be altered.

Driving the time-critical software by interrupt is yet another possibility. In the present routine, precious micro-seconds are wasted as software scans for the end-of-conversion signal, and this reduces the time available for packing data into bytes. The E-O-C signal can trigger the non-maskable-interrupt to bring in immediately a data packing routine, though sadly a bug in the current ROM prevents this particular experiment on a Spectrum.

E&CM



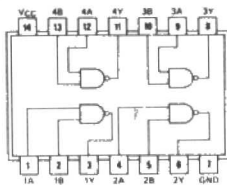
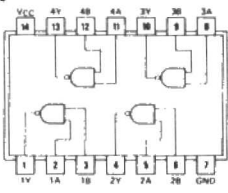
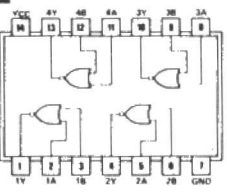
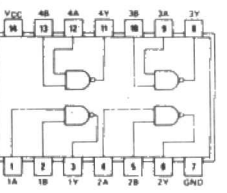
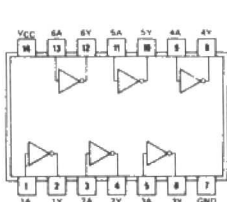
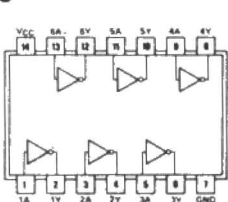
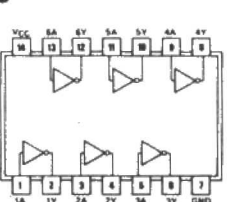
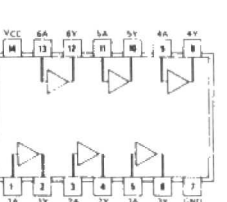
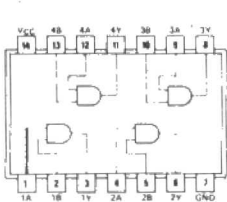
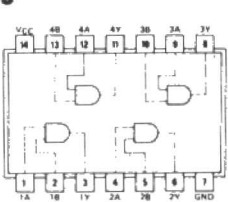
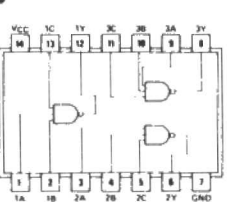
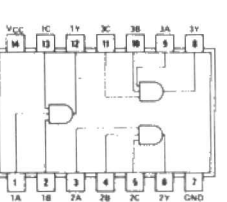
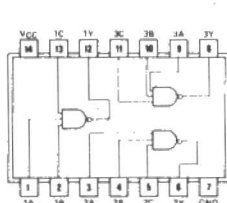
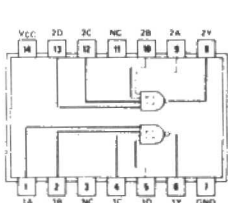
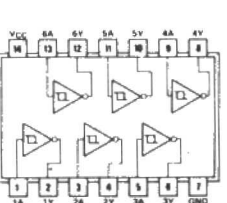
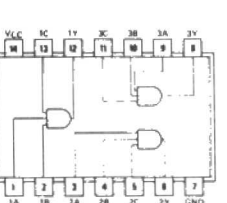
Three sample outputs showing (top) an apple, (centre) a glass and spoon and (above), our logo.

INTEGRATED DATA

You will be forgiven if these pages promote a feeling of *de ja vu* as these Integrated Data pages look much the same as last month's. A number of gremlins crept into the pro-

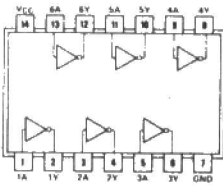
duction of June's feature however and meant that the majority of devices were incorrectly identified. Sack cloth and ashes were the order of the month in the

proofing department after the event – they assure us however that everything is in order this time. More details of TTL pin-outs next month.

<p>00</p>  <p>QUADRUPLE 2-INPUT POSITIVE-NAND GATES</p>	<p>01</p>  <p>QUADRUPLE 2-INPUT POSITIVE-NAND GATES WITH OPEN-COLLECTOR OUTPUTS</p>	<p>02</p>  <p>QUADRUPLE 2-INPUT POSITIVE-NOR GATES</p>	<p>03</p>  <p>QUADRUPLE 2-INPUT POSITIVE-NAND GATES WITH OPEN-COLLECTOR OUTPUTS</p>
<p>04</p>  <p>HEX INVERTERS</p>	<p>05</p>  <p>HEX INVERTERS WITH OPEN-COLLECTOR OUTPUTS</p>	<p>06</p>  <p>HEX INVERTER BUFFERS/DRIVERS WITH OPEN-COLLECTOR HIGH-VOLTAGE OUTPUTS</p>	<p>07</p>  <p>HEX BUFFERS/DRIVERS WITH OPEN-COLLECTOR HIGH-VOLTAGE OUTPUTS</p>
<p>08</p>  <p>QUADRUPLE 2-INPUT POSITIVE-AND GATES</p>	<p>09</p>  <p>QUADRUPLE 2-INPUT POSITIVE-AND GATES WITH OPEN-COLLECTOR OUTPUTS</p>	<p>10</p>  <p>TRIPLE 3-INPUT POSITIVE-NAND GATES</p>	<p>11</p>  <p>TRIPLE 3-INPUT POSITIVE-AND GATES</p>
<p>12</p>  <p>TRIPLE 3-INPUT POSITIVE-NAND GATES WITH OPEN-COLLECTOR OUTPUTS</p>	<p>13</p>  <p>DUAL 4-INPUT POSITIVE-NAND SCHMITT TRIGGERS</p>	<p>14</p>  <p>HEX SCHMITT-TRIGGER INVERTERS</p>	<p>15</p>  <p>TRIPLE 3-INPUT POSITIVE-AND GATES WITH OPEN-COLLECTOR OUTPUTS</p>

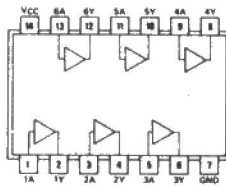
INTEGRATED DATA

16



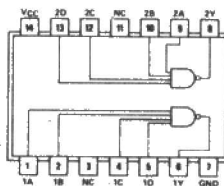
HEX INVERTER BUFFERS/DRIVERS
WITH OPEN-COLLECTOR
HIGH-VOLTAGE OUTPUTS

17



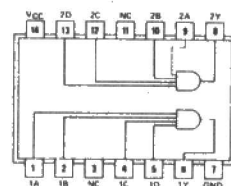
HEX BUFFERS/DRIVERS
WITH OPEN-COLLECTOR
HIGH-VOLTAGE OUTPUTS

20



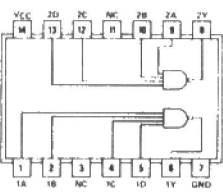
DUAL 4-INPUT
POSITIVE-NAND GATES

21



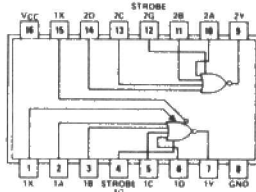
DUAL 4-INPUT
POSITIVE-AND GATES

22



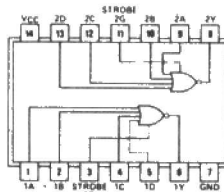
DUAL 4-INPUT
POSITIVE-NAND GATES
WITH OPEN-COLLECTOR OUTPUTS

23



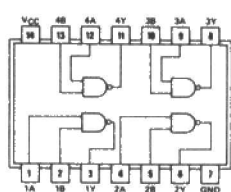
EXPANDABLE DUAL 4-INPUT
POSITIVE-NOR GATES
WITH STROBE

25



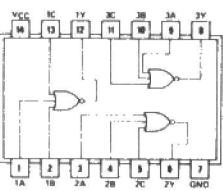
DUAL 4-INPUT
POSITIVE-NOR GATES
WITH STROBE

26



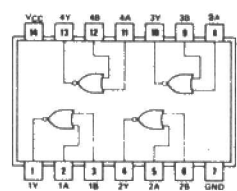
QUADRUPLE 2-INPUT
HIGH-VOLTAGE INTERFACE
POSITIVE-NAND GATES

27



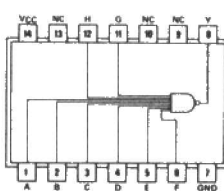
TRIPLE 3-INPUT
POSITIVE-NOR GATES

28



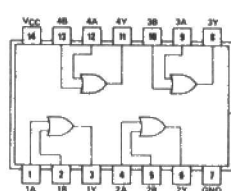
QUADRUPLE 2-INPUT
POSITIVE-NOR BUFFERS

30



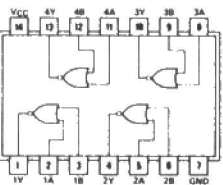
8-INPUT
POSITIVE-NAND GATES

32



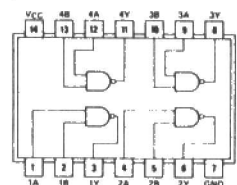
QUADRUPLE 2-INPUT
POSITIVE-OR GATES

33



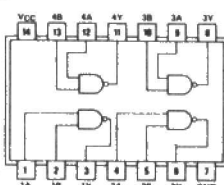
QUADRUPLE 2-INPUT
POSITIVE-NOR BUFFERS
WITH OPEN-COLLECTOR OUTPUTS

37



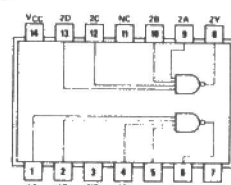
QUADRUPLE 2-INPUT
POSITIVE-NAND BUFFERS

38



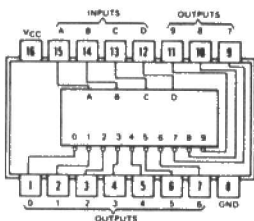
QUADRUPLE 2-INPUT
POSITIVE-NAND BUFFERS
WITH OPEN-COLLECTOR OUTPUTS

40



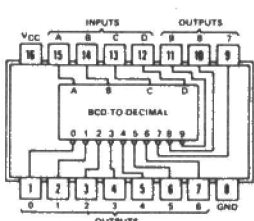
DUAL 4-INPUT
POSITIVE-NAND BUFFERS

42 43 44



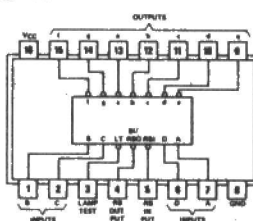
4 LINE-TO-10-LINE DECODERS

45



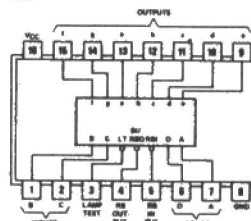
BCD-TO-DECIMAL DECODER/DRIVER

46 47




BCD-TO-SEVEN-SEGMENT
DECODERS/DRIVERS

48



BCD-TO-SEVEN-SEGMENT
DECODERS/DRIVERS



"THERE MUST BE A COMPUTER DEALER I CAN TURN TO FOR GUIDANCE...."

If you're still staggering through the computer jungle and not getting sensible answers to your questions, we have some good news:

Now you can turn to **professional** people who are capable of giving you sound advice on practically all aspects of popular computing.

They all have one thing in common:

They are **COMPUTERS FOR ALL** dealers.

A Computers for All dealer is different from the normal Computer retailer.

Not surprisingly he won't try to sell you things like cameras or cosmetics, stationery or sealing wax.

He will, however, be capable of answering sensibly almost any question you have on computers and computing, and have readily available a wide range of popular computers, hardware, software, books, and peripherals.

So why not call in at your local **COMPUTERS FOR ALL** dealer today?

He can lead you in the direction you want to travel.

The shops where people matter:

AVON
JADE COMPUTERS
Coombend, Radstock, Bath.
0761 32570.

***MERCATOR COMPUTER SYSTEMS**
3 Whiteladies Road, Clifton, Bristol. 0272 731079.

MOBILE MICROS
2 Castle Street, Thornbury, Bristol. 0454 418383.

BERKSHIRE
***KENNETH WARD COMPUTERS**
Verve House, London Road, Sunningdale. 0990 25025.

CHANNEL ISLANDS
MEGA LTD.
7 Anley Street, St. Helier, Jersey. 0534 72263.

CORNWALL
***FAL-SOFT COMPUTERS**
8 St. George's Arcade, Falmouth. 0326 314663.

DEVON
***BITS & BYTES**
44 Fore Street, Ilfracombe, N. Devon EX34 9JD. 0271 62801.

COMPUTER SYSTEMS (TORBAY)
Pump Street, Brixham. 08045 6565/6.

***CRYSTAL COMPUTERS**
209 Union Street, Torquay. 0803 22699.

DORSET
DENSHAM COMPUTERS
329 Ashley Road, Parkstone, Poole. 0202 737493.

***WEYTECH COMPUTER SYSTEMS**
20 St. Edmunds Street, Weymouth. 03057 79881.

ESSEX
***AKHTER INSTRUMENTS**
Unit 19, Airlinghyde Estates, South Road, Harlow. 0279 412639.

***COMPUTERS FOR ALL**
72 North Street, Romford. 0708 752862.

***EMPRISE LTD.**
58 East Street, Colchester. 0206 870353.

***EMPRISE LTD.**
3a Baddow Road, Chelmsford. 0245 356834.

HANTS
HOME COMPUTER CENTRE (ROSS RECORDS)
13 Kingston Road, Portsmouth. 0703 819515.

MICRO VIDEO STUDIOS LTD.
17 Turks Street, Alton. 0420 82055.

HERTS
VIDEO CITY COMPUTERS
45-47 Fishers Green Road, Stevenage. 0438 53808.

KENT
***APHROS SOFTWARE CO.**
83 Canterbury Road, Westbrook, Margate. 0843 23627.

***MEDWAY COMPUTERS**
141 New Road, Chatham. 0634 826080.

***MICRO MAGIC**
128 Erith Road, Bexleyheath. 0322 523052.

LEICS
DIMENSION
27-29 High Street, Leicester. 0533 57479.

LONDON
EUROCALC
224 Tottenham Court Rd., London W1. 01-631 4139.

KAYDE HOME COMPUTERS
1 Station Approach, New Eltham, London SE9. 01-859 7505.

***KELLY'S COMPUTERMARKET**
227 Dartmouth Road, Sydenham, London SE26 4QY. 01-699 4399/6202.

MIDDLESEX
MAYFAIR COMPUTER SERVICES
6 Main Avenue, Moor Park, Northwood. 09274 20664.

***TWILLSTAR COMPUTERS**
17 Regina Road, Southall. 01-574 5271.

N. IRELAND
D. V. MARTIN LTD.
13 Bridge Street, Belfast. BT1 1LT. 0232 226434.

NORFOLK
***CARLTON COMPUTERS**
4 Swanstons Road, Great Yarmouth, NR30 3NQ. 0493 58898.

S. WALES
***TOMORROW'S WORLD**
Esplanade, Lerwick, ZE1 0LL. 0595 2145.

SUFFOLK
BECCLES COMPUTERS
5 The Ridings, Worlingham, Beccles. 0502 715061.

SURREY
***ANIROG COMPUTERS**
8 High Street, Horley. 02934 2007.

COMPUTASOLVE
8 Central Parade, St. Marks Hill, Surbiton. 01-390 5135.

SUSSEX
THE COMPUTER CENTRE (BMS) LTD.
37d & 37e Robertson Street, Hastings, East Sussex. 0424 439190.

S. WALES
AUTOMATION SERVICES (S. WALES)
3 Werrneys Road, Penystai, Bridgend. 0656 720959.

DAN EVANS (BARRY) LTD.
81 Holton Road, Barry, South Glamorgan. 0446 734242.

MICRO-CARE COMPUTING
18 Hawarden Road, Newport, Gwent. 0633 278040.

***MORRISTON COMPUTER CENTRE**
37 Clase Road, Morriston, Swansea SA6 8DS. 0792 797572.

STEVE'S COMPUTER CO. LTD.
Castle Arcade, Cardiff. 0222 41905.

WARWICKSHIRE
***IMPULSE MICRO SYSTEMS LTD.**
6 Central Chambers, Cooks Alley, Wood Street, Stratford-upon-Avon. 0789 295819.

WEST MIDLANDS
JBC
200 Earlsdon Ave, North Earlsdon, Coventry. 0203 73813.

WORCESTERSHIRE
***DEATH VALLEY COMPUTERS**
P.O. Box 54, Worcester WR2 6QA. 0905 640400.

EVESHAM COMPUTER CENTRE
Crown Court Yard, Bridge St., Evesham. 0386 48635.



COMPUTERS FOR ALL